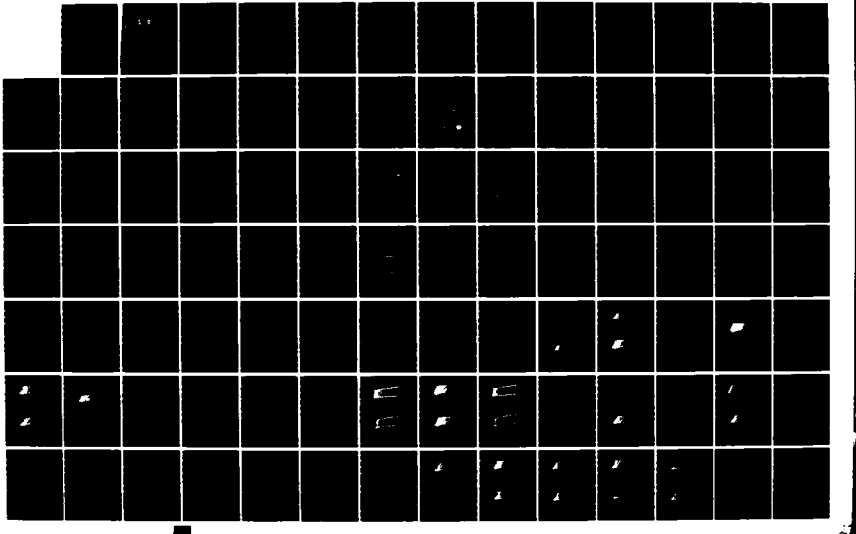
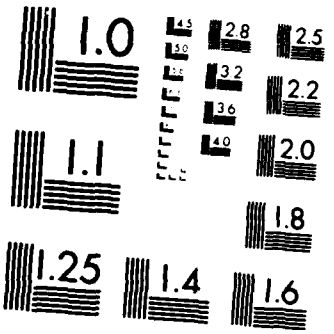
AD-A173 546 COMPUTER GRAPHICS FOR SYSTEM EFFECTIVENESS ANALYSIS(U) 1/2
MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR
INFORMATION AND DECISION SYSTEMS C M BOHNER MAY 86
UNCLASSIFIED LIDS-TH-1573 N00014-84-K-0519 F/G 9/2 NL





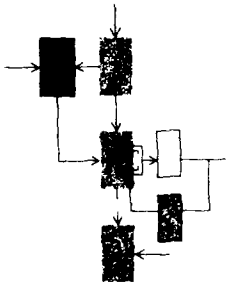
JULY 1986

LIDS-TH-1573

Research Supported By:

Office of Naval Research
Contract No. N00014-84-K-0519

Joint Directors of Laboratories
through the Office of Naval Research
Contract No. N00014-85-K-0782



COMPUTER GRAPHICS FOR SYSTEM EFFECTIVENESS ANALYSIS

Christine M. Bohner

Laboratory for Information and Decision Systems
MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

MAY 1986 LIDS-TH-1573

COMPUTER GRAPHICS

FOR

SYSTEM EFFECTIVENESS ANALYSIS

by

Christine M. Bohner

This report is based on the unaltered thesis of Christine M. Bohner, submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Technology and Policy at the Massachusetts Institute of Technology in May 1986. The research was conducted at the MIT Laboratory for Information and Decision Systems with support provided in part by the Office of Naval Research under Contract No. N00014-84-K-0519 and by the Joint Directors of Laboratories through the Office of Naval Research under Contract No. N00014-85-K-0782.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Computer Graphics

for

System Effectiveness Analysis

by
Christine M. Bohner
(1986)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE
IN
TECHNOLOGY AND POLICY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1986

Signature of Author _____
Department of Electrical Engineering and Computer Science
May 1986

Certified by _____
Alexander H. Levis
Thesis Supervisor

Accepted by _____
Richard de Neufville
Chairman, Technology and Policy Program

Accepted by _____
Arthur C. Smith
Chairman, Departmental Graduate Committee
Electrical Engineering and Computer Science Department

Computer Graphics

for

System Effectiveness Analysis

by

Christine M. Bohner

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 1986
in partial fulfillment of the requirements for the degree of
Master of Science in Technology and Policy

ABSTRACT

Systems Effectiveness Analysis (SEA) is a methodology for assessing the effectiveness of a system by constructing and evaluating surfaces that represent the performance characteristics of the system and the requirements of the task to be performed. A computer graphics program has been designed and implemented on a workstation and is dedicated to constructing the three dimensional projection of higher dimensional surfaces or loci. The program allows the user to view the loci in different dimensions which serves as an aid in analysis. To demonstrate the use of the program, the effectiveness of a C_{11}^{3} system is analyzed. Two aspects of the system are studied: the effect of adding more components to the system, and the sensitivity of the overall effectiveness to different errors within the measuring equipment of the system. The graphics system is used to generate the plots of the system loci and to evaluate its performance.

Thesis Supervisor: Dr. Alexander H. Levis

Title: Senior Research Scientist

ACKNOWLEDGEMENT

I would like to thank Dr. Alexander H. Levis for his time and patience in helping me complete this thesis. His guidance and support has made my thesis the most valuable part of my education. His concern is greatly appreciated.

I would also like to thank my research associates Scott, Vicky, Ashima, John, Anne-Claire, Muriel, Stamos, Pascal, Phillipe, and Herve. There help and cooperation made working on the thesis more enjoyable. Lisa Babine has also been very supportive of this work.

The support of my best of friends Hiroshi, Howie, Diana, and Maura during the difficult moments of my work is something I am greatly thankful for.....

Finally, for my family, who has helped me through school in many ways, I am most thankful...

This research was conducted at the MIT Laboratory for Information and Decision Systems with support provided in part by the Office of Naval Research under Contract No. N00014-84-K-0519 and by the Joint Directors of Laboratories under Contract No. N00014-85-K-0782.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Introduction

Computer graphics is being used in many different disciplines of engineering and science to aid both the engineer in developing systems and the scientisit in developing models. Computer hardware has decreased in cost, and today, it is feasible for a small group of scientists and engineers to have a personal computer workstation to aid in design. Engineering design has been transformed from two-dimensional paper-and-pencil work to three-dimensional computer-aided design. The computer graphics program developed in this thesis is one tool for such an engineering workstation.

With the aid of the graphics system, research in system effectiveness analysis (SEA) was completed for this thesis. SEA, a methodological framework [Bouthonnier, 1982], has been used to evaluate the effectivess of C^3 systems[Bouthonnier, 1982][Cothier, 1984][Karam, 1985], manufacturing systems [Washington, 1985], automotive systems[Levis, Houpt, and Andreadakis, 1984], and power systems [Dersin and Levis, 1982]. This methodology is used to analyze loci of a system and a task given to that system in a certain context.. From this analysis, an assessment of how well the system performs under the given task can be done. Other system properties, such as the effect of time on the system performance (Cothier, 1984), or sensitivity of the system performance to errors within its components, can be studied with the use of the graphics system. Most of the loci generated from earlier research had been plotted by hand, which was tedious and imprecise. With the graphics system, these loci can be constructed within seconds and viewed in different two and three-dimensional projections. This allows a researcher to view changes to a system quickly and precisely.

1.2 Outline of the thesis

This thesis is organized into seven chapters as follows. Chapter II gives a detailed description of the methodological framework of system effectiveness analysis. It describes how the mission is evaluated under a given task within the context in which the system operates. The graphics system is described in Chapter III, and how it generates surfaces is

detailed. The graphics system is demonstrated by applying the system effectiveness analysis to the TACFIRE system. The mathematical model of this system is derived in Chapter IV. Chapters V and VI present the results of the analysis. Chapter VII summarizes the results and gives recommendations for future research.

CHAPTER II

METHOD OF ANALYSIS

2.1 Introduction

Systems effectiveness analysis (SEA) is a methodology for measuring how well a system accomplishes a given task. This methodology allows for both qualitative and quantatative analysis of a system's effectiveness. The power of the methodology is that it can be applied to many different system/task configurations. It was first used on power systems by Dersin and Levis (1981,1982). Bouthonnier and Levis (1984) and Cothier and Levis (1984) furthered the methodology by applying it to military Command, Control, and Communication (C^3) systems, and Karam (1985) continued this analysis with evolving systems. In mechanical systems, Levis, Houpt, and Andreadakis (1984) applied the methodology to determine the effectiveness of an automotive system, while Washington (1985) assessed the effectiveness of alternative flexible manufacturing systems. The description of the methodology presented here closely follows the description in the earlier research cited above.

2.2 System Effectiveness Analysis (SEA)

The key notion in SEA is to measure how effective a system is in meeting the requirements of an assigned task, with the system and the task in the same context. To present the methodology, it is necessary to define some terms.

System: The system is the interconnection of technologies and equipment working under a standard operating procedure. A system could be a military C^3 system, an automotive system, a large-scale power system, or a flexible-manufacturing system (FMS).

Mission: Simply the task given to the system. For example, a mission for a C^3 system may be the task assigned to the military unit or ogranization.

Context: The context is the environment in which the system operates and the mission is required to take place. The context can include geographical location, weather conditions

and/or a network configuration.

Primitives: The primitives are the basic characteristics of the system and the mission in the given context. In the analysis, they are considered to be the independent variables. Let $\{x_i\}$ be the set of system primitives and $\{y_i\}$ be the set of mission primitives.

Attributes: Attributes describe the properties of the system that are revelant to the mission. Let $\{A_s\}$ denote the set of systems attributes and $\{A_m\}$ denote the set of mission attributes. Attributes are functions of the primitives. Attributes are also called Measures of Performance (MOPs).

Measures of Effectiveness (MOE's): Measures of Effectiveness are measures of how well the system performs the tasks in a given context. They show how well the system meets the mission requirements.

The system, mission, task, and context should not be defined broadly, but should be described precisely. For example, a system described as a "communications system of voice, telephone, and computer links" is not defined concretely enough for analysis. The same system, with specifications that include maps of locations of the components and interconnections, and statistics of the length of time for messages to travel, would be necessary.

Once the system and mission are defined, and primitives are specified, attributes that describe the system and mission can be derived. Attributes can be functions of more than one primitive; therefore, change in any one of the primitives may change more than one attribute.

$$A_s = f(x_1, x_2, \ldots) \qquad s=1,2,\ldots \qquad (2.1)$$

$$A_m = f(x_1, x_2, \ldots) \qquad m=1,2,\ldots \qquad (2.2)$$

2.3 Analysis of Performance Loci

As the primitives vary, the attributes also vary correspondingly. The attributes must be mapped to an N-dimensional space where each dimension represents a different attribute. As the system primitives vary between their minimum and maximum values, the system attributes (which are functions of the primitives) sweep out a locus. Similarly, a locus can be defined by the mission attributes.

From the relative geometry of the loci, a measure of effectiveness can be computed. Let L_s denote the system locus and L_m denote the mission locus. The geometry can be classified as two cases:

$$L_s \cap L_m = 0 \tag{2.3}$$

The system does not satisfy the mission at all.
Alternatively,

$$L_s \cap L_m \neq 0 \tag{2.4}$$

with

$$L_s \cap L_m < L_s$$

and

$$L_s \cap L_m < L_m$$

The mission is partially satisfied by the system.
There are two special cases to consider:

$$L_s \cap L_m = L_s \tag{2.5}$$

The system locus is entirely contained in the mission locus. This is the most effective configuration because the system completely satisfies the mission.

$$L_s \cap L_m = L_m \tag{2.6}$$

The mission locus is inside the system locus. This indicates that the system is capable of fufilling the mission. However, the system is not being utilized effectively because it operates in ranges which do not satisfy the mission.

The measurement described above is one of many different measures which describe the extent to which the system meets the mission requirements. Each of these measures of effectiveness can be normalized to take on the values between 0 and 1. Let $\wp$ denote a measure in normalized attribute space. One partial measure of effectiveness is

$$E_p = \wp(L_s \cap L_r) / \wp(L_s) \qquad (2.8)$$

which assigns a numerical value to how well the system satisfies the requirements given to it.

This, however is one in the set of all partial measures $\{E_p\}$. They are combined to form a global measure with the use of a utility function. This allows for judgement to be included in the methodology. Utility theory enables one to "weight" the partial MOE's with the following function:

$$E = u(E_1, E_2, \ldots, E_k) \qquad (2.9)$$

The methodology is shown in Figure 2.1. Note that the mission and the system locus are derived independently.

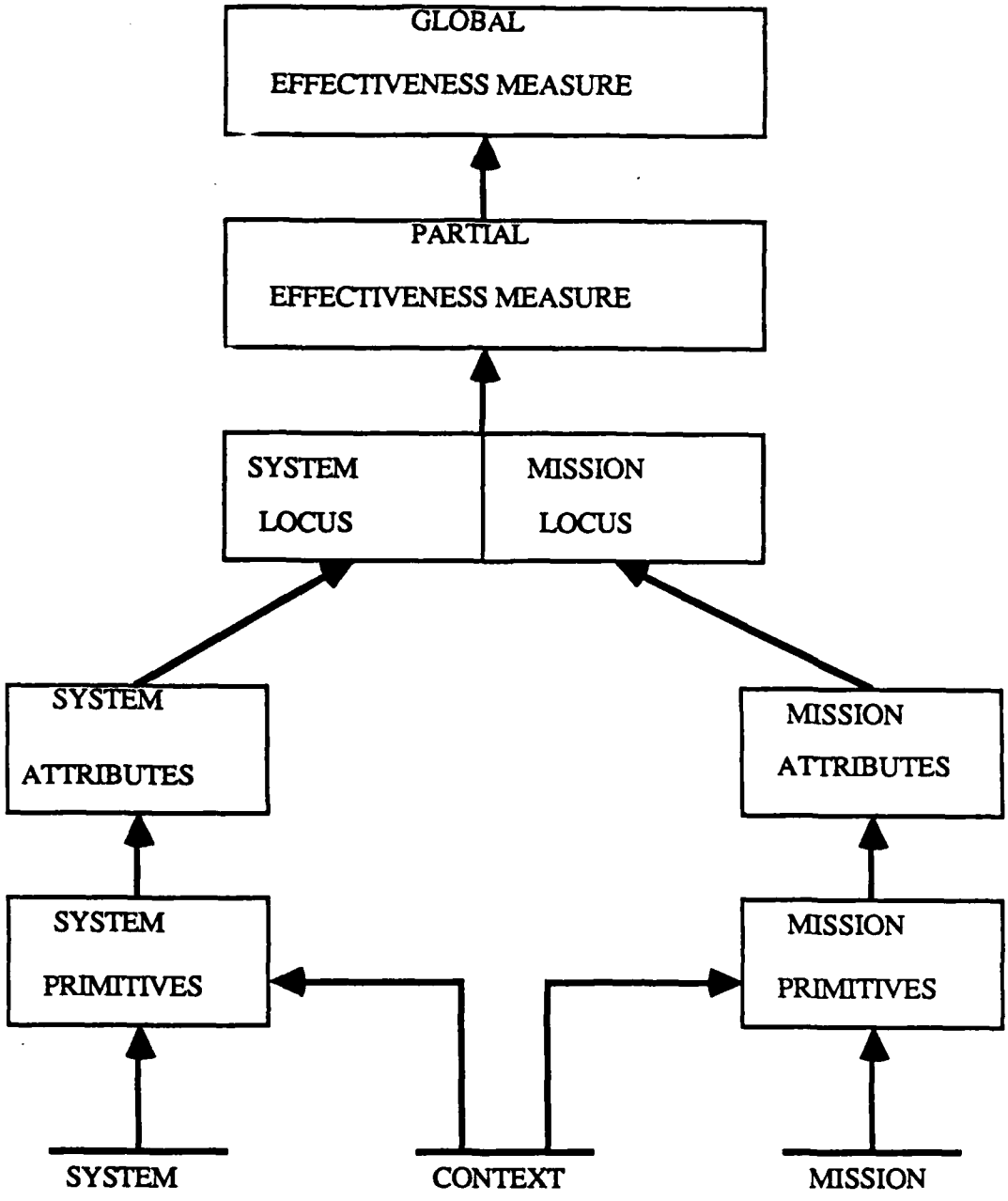


Figure 2.1 The Methodology of System Effectiveness Analysis

CHAPTER III

THE GRAPHICS SOFTWARE

3.1 Introduction

With the wide use and low prices of microcomputer systems today, many scientists and engineers are able to build their own computer workstations to address their specific needs. The first step is obtaining the necessary hardware - computers, graphics hardware, displays, printers, and plotters. The next step is designing and implementing the software to perform the workstation's tasks. The last step is designing a user interface, an important feature for the workstation to be useful. This section will describe:

- The workstation hardware and software.
- The algorithms used in the software design.
- Implementation of the software.
- The user interface.
- How this workstation can be used to support the systems effectiveness analysis methodology.

3.2 Software/Hardware System

The hardware and software of the workstation combine to form the system shown in Figure 3.1. The key words in this figure are:

GDT: Graphics Development Toolkit.

PGC: Professional Graphics Controller.

VDI: Virtual Device Interface.

Refer to Appendix A for hardware and software specifications.

The flow of information is displayed in Figure 3.1. The user chooses which data file to read into the data structure and then chooses a viewing configuration. If two-dimensional viewing is chosen, the Graphics Development Toolkit (GDT) routines are used to plot the data for different devices, such as the display, the printer, or the plotter. Data is sent through the Virtual Device Interface (VDI). The VDI is the result of efforts to standardize computer

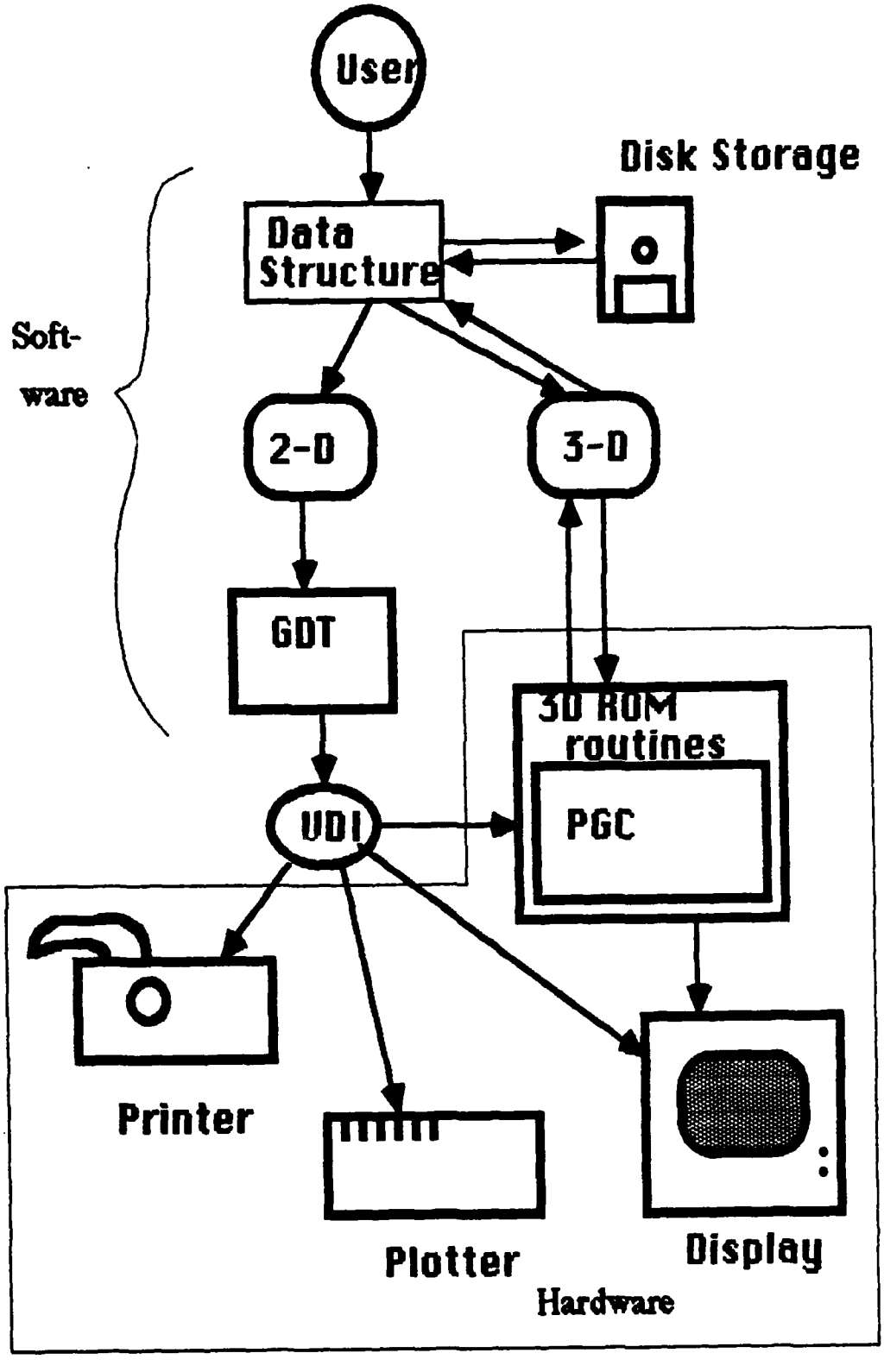


Figure 3.1 Hardware and Software of the Engineering Workstation

graphics by making graphic pictures device-independent. Thus, graphical data can be sent to the VDI, and then to specific device drivers. These drivers translate the graphical data into aform that the particular device can read. If three-dimensional viewing is chosen, graphics routines are used to plot the data to the display. These graphics routines (called high-function graphic routines by IBM) are in the read-only memory (ROM) hardware on the Professional Graphics Controller (PGC) card. Commands are sent to the card with an assembly-language subroutine. Once the ROM routines are executed, the data is sent to the display directly. Graphics produced in this way depend on the PGC, and thus are device-dependent. To make these pictures device-independent, transformed data is read back from the card and saved in a disk file. This data can then be viewed in 2 dimensions and sent through the VDI for plotting, printing, or displaying.

3.3 The Data Structure

The data structure serves as the functional base of the software in the engineering workstation. The data structure in this workstation is designed for simple construction of the loci and has the form:

LOCI[i, j, k, l | x_1,x_2,....,x_7]

i,j,k,l: problem dependent data (see example below).

{x_1,.....,x_n} : physical points in n-dimensional space.

This data structure is flexible enough to handle locus data from the three systems to which the systems effectiveness methodology has been applied - automotive, C^3, manufacturing; and even decision-making organizational systems, where a different methodology is used to generate the data for the loci.

All of the data for the loci will be generated independently of the software package and will be stored in a file on the hard disk or a floppy disk. A programmer will have knowledge of this data structure and should store data to a file in a standardized manner. The data must be stored to a file on a disk with the following rules:

1. **The file must be a formatted file** (that is, it must have carriage returns at the end of each line).
2. **The first line of the file will contain the maximum number of indices for the i, j, k, and l index. This will have the order:**

   IMAX JMAX KMAX LMAX
3. **The rest of the file will contain just data numbers which must be written as in standard format, NOT scientific notation. For example, the number 12.34567 must be written to the file as such, not as 0.1234567 E02.**
4. **Each line will have the form:**

   I J K L x_1 x_2 x_3 x_4 x_5 x_6 x_7

   At least one space must be between each data entry and no lines should be skipped between lines of data.

A procedure will read the file of this format and assign the data to the data structure.

Two main aspects of the data structure are as follows:

• The four indices i, j, k, l are problem dependent. That is, the programmer or software which creates the data file will assign values to i, j, k and l. For example, i, j, k, and l can be parameters of the function which generates the locus - F(i,j,k,l). The function may in fact only use one two or three of these indices as parameters. With each combination of parameters, a point is computed from the function. Varying theses parameters over their minimum and maximum values will generate a locus. Let f(i,j,k) be a locus function. The index l is not a parameter and therefore has no values assinged to it. The locus will be viewed in three dimensions and if it consists of more than three, the viewer will choose which three dimensional projection to view. Note that the indices do not usually correspond to the dimensions. If the locus -function is derived from the SEA methodology, it is typically a function of other functions. Let $g_1(i,j,k)$, $g_2(i,j,k)$, $g_3(i,j,k)$ be functions of the indices. The locus-function is then $f(g_1,g_2,g_3)$.

The physical points generated by the locus-function are assigned to a data structure or file in the order the indices or parameters were varied during computation. The power of this structure is that the user is able to choose indices **i, j, k** and **l** to control how data points are displayed. This concept is illustrated by the following example.

The user selects which three of of the seven dimensions to view and how the indices will vary. In Figure 3.2, the user has chosen **j** to vary first and then **i** to vary second. Index **j** is increasing in the direction of the arrow. Thus, **j** varies as **i** is constant; then **i** is increased and **j** varies again over the same range. The index **k** is held constant during both these variations. Each of the data points is represented by a black circle.

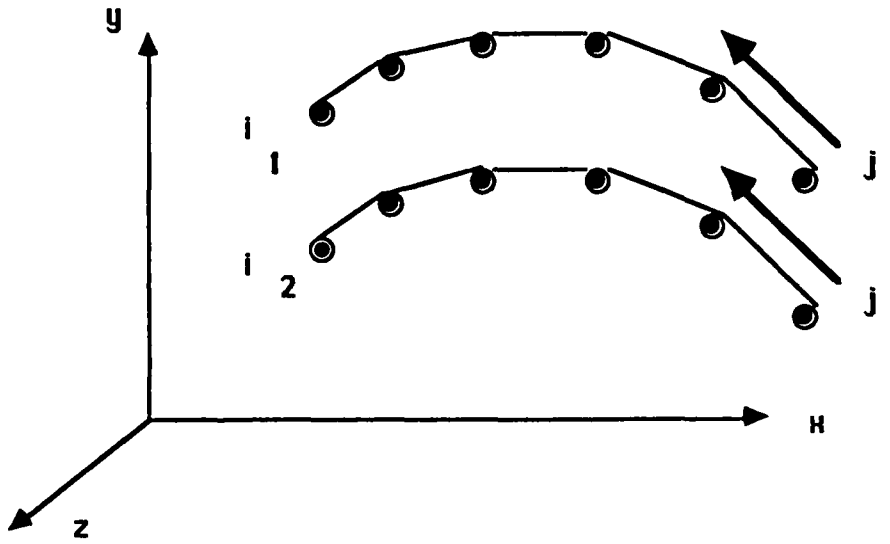


Figure 3.2 Variation of the indices i and j of a locus

In Figure 3.2, **j** is classified as a **major** (primary) index and **i** is a **minor** (secondary) index. In Figure 3.3, **k** is varied while **j** is held constant, and then **j** is changed, and **k** is varied again. The index **i** is held constant. Notice that the data points are the same for each variation, they are just connected by different lines as the indices are changed.

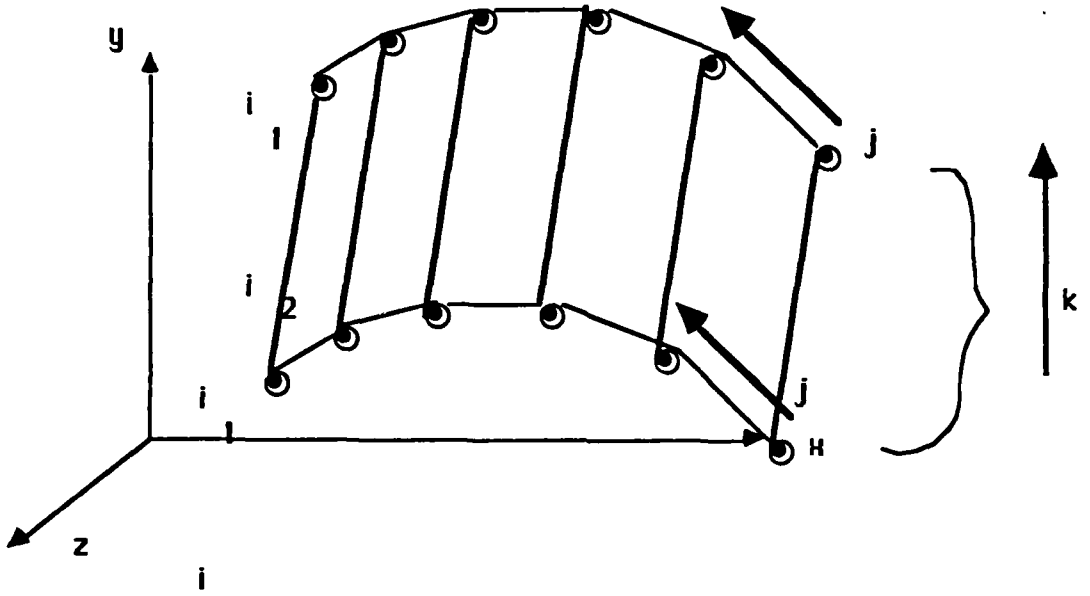


Figure 3.3 Variation of the indices k and j of a locus

Now if all the points are connected with all of the variations, a wireframe mesh of the locus results as shown in Figure 3.4.

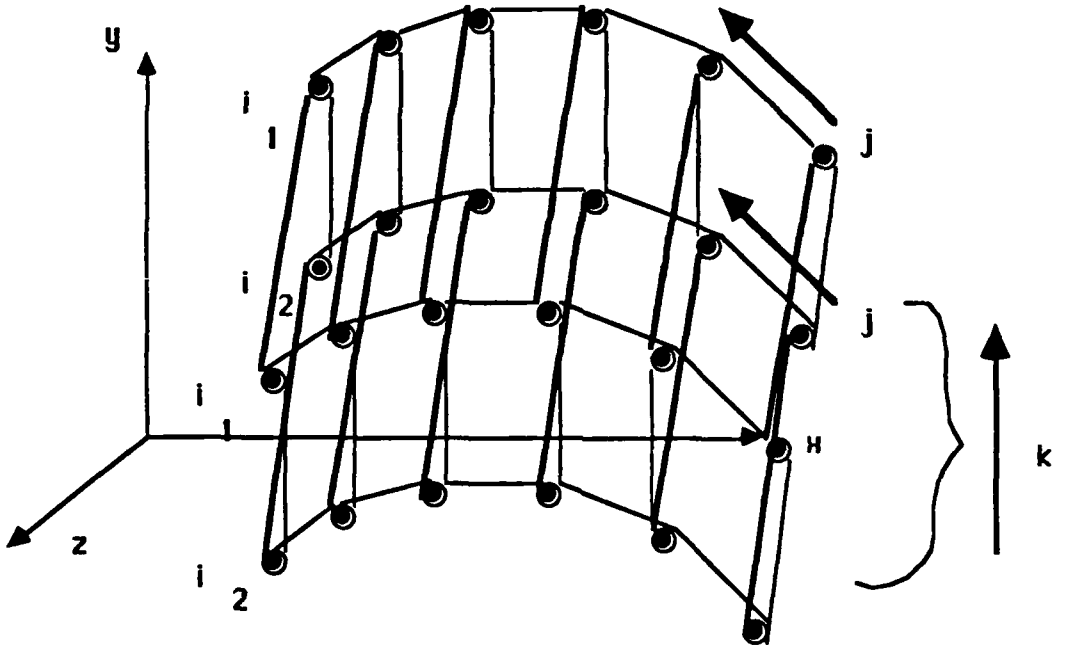


Figure 3.4 Index variation of i, j, and k for a locus

Moreover, we can generate a wire frame by using the first and last planes in each set of parallel cross-sections as shown in Figure 3.5

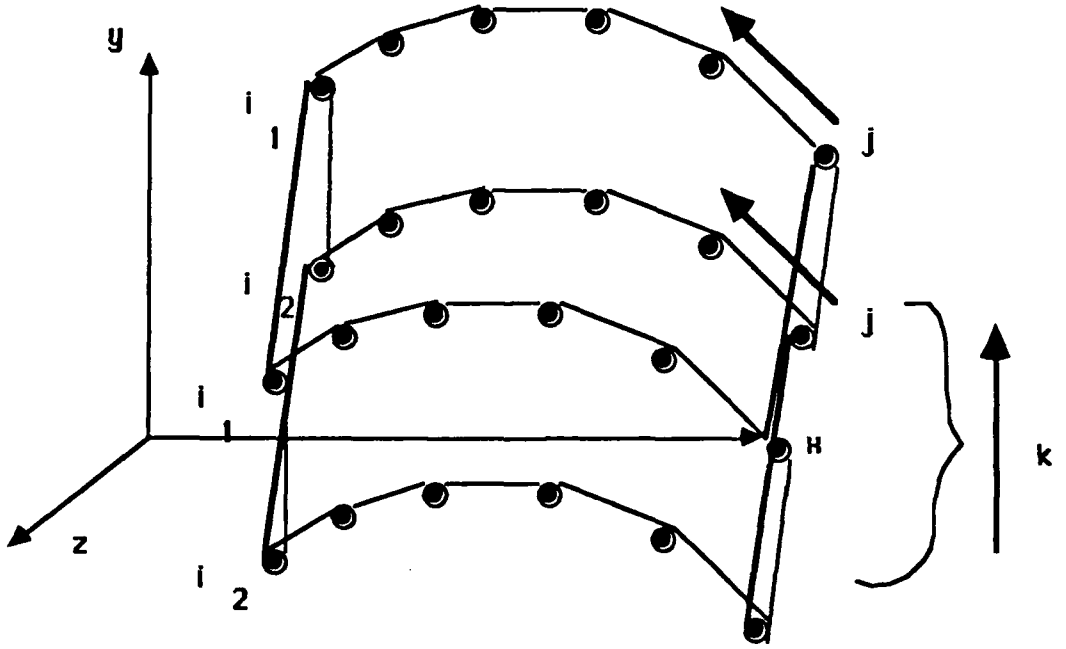


Figure 3.5 Wireframe Representation of Locus

This general data structure is useful because it allows for different ways of viewing the same locus.

3.4 Algorithms

Once the data is assigned to the structure, the user selects the order of indices to plot the data. Line drawing is done by calling the line drawing routines in the Graphics Development Toolkit for two-dimensional viewing, or in the ROM of the PGC for three-dimensional viewing. But, before this can be done, data must be mapped from wherever it is in the three -dimensional space to the two-dimensional screen.

3.4.1 Mapping from world coordinates to screen coordinates

Figure 3.6 illustrates how the data mapping is done.

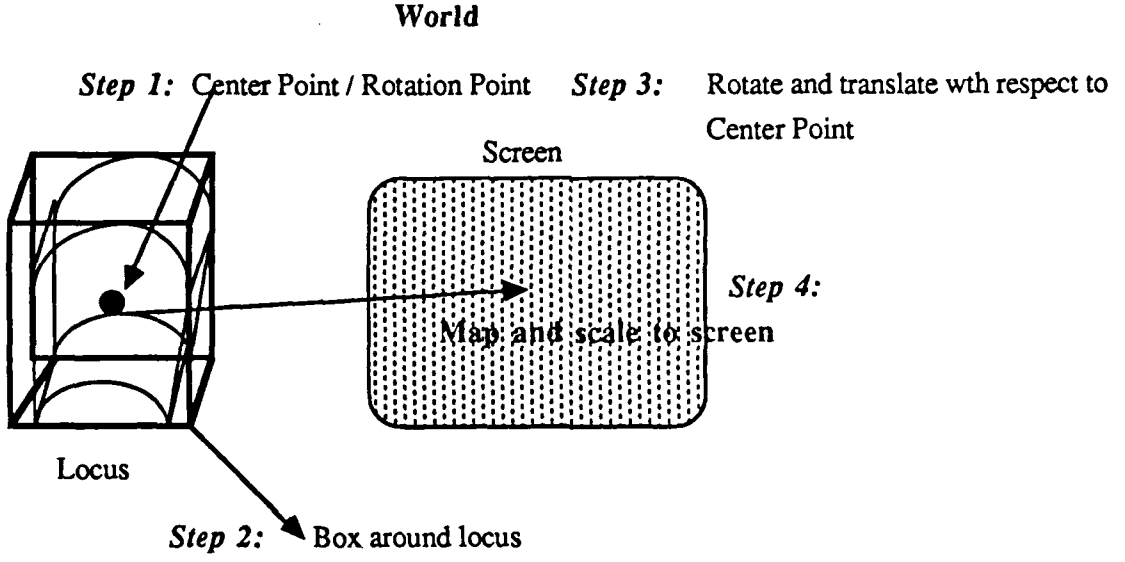


Figure 3.6 Mapping from world coordinates to the screen

The algorithm is:

- **step 1:** Compute the center of the object in n-space.
- **step 2:** Form a cube around the object. Map and scale to screen coordinates.
- **step 3:** Rotation or translation is executed if requested. Rotation computations are executed with respect to the center of the object in 3-D (the three dimensions chosen by user to view).
- **step 4:** Orthographically project object to screen.

The algothrithm is the same for two-dimensions except the z coordinate is not considered and step 4 is not needed.

Most of the three-dimensional computations, such as rotation, mapping to the

screen coordinates, and translation, are executed by the PGC. The PGC contains an 8088 microprocessor, which is the graphics controller. The 80286 system microprocessor communicates with the 8088 through a data bus and an address bus. Besides the commands the system sends, a program can read and write data to memory locations, which are then sent to the PGC interface, along with the bus data. This interface interprets only recongnizable commands and sends them to the 8088. To call the ROM routines, a program sends *high-function graphics* commands to the write-memory locations. The 8088 will interpret those commands with the ROM into data and write the data to the 64K of display memory on the PGC. The contents of this memory are sent to the screen display. The commands are written to the memory locations with assembly language routines. Data from the display memory can also be read back to the system with assembly language routines. (See Appendix B for complete details on these routines.) For specific details on the 3-D algorithms used by the PGC, refer to the IBM Personal Computer Professional Graphics Controller Technical Reference , Personal Computer Hardware Reference Library, IBM Corp, Inc., August 15, 1984.

3.5 Implementation

Developing the algorithms in software design is only one aspect of the process. Implementation of these algorithms into software can be the most difficult aspect of the design. In the software design, there were many implementation problems, but only two will be discussed: the data structure design in FORTRAN, and sending high-function graphics commands to the ROM of the Professional Graphics Controller (PGC). These two aspects are crucial to understanding how the software works.

3.5.1 The Data Structure implemented in FORTRAN

The data structure can be easily implemented as a pointer-record structure. In this record, a unique pointer is assigned to each data point and index combination. In FORTRAN, this is implemented in the following way:

```
pointer(max)
loci(pointer(max),7)
index(pointer(max),4)
```

The pointer values are determined by the index values according to the formula:

```
m=0
 for ibar=1,imax
   for jbar=1,jmax
     for kbar=1,kmax
          for lbar=1,lmax
         m=m+1
           pointer(m)=i + 10*(j-1) + 100*(k-1) + 1000*(l-1)
```

For each **m** there is a unique pointer that is an array index for the arrays **loci** and **index**. This is demonstated by the following numberical example:

i=2 j=5 k=10 l=1

pointer(m_1) = 2 + 10*4 + 100*9 +0= 942

If k increase to k=3,the pointer becomes

pointer(m_2) = 943

This formula will fail for **imax, jmax, kmax,** or **lmax** greater than 10. For example, if **i=2, j=12, k=1, and l=1 then pointer(m_1) = 112** but if **i=2,j=2,k=2** then **pointer(m_2) = 112** and thus the pointer is not unique. This problem is solved by using the following adjusted pointer equation.

pointer(m) = i*imax + 10*jmax*(j-1) + 100*kmax*(k-1) + 1000*lmax*(l-1)

This will assure that each pointer number will be unique for each variation of indices.

3.5.2 Communications with the PGC ROM

Communications with the PGC card follows a protocol designed by IBM©. High-function graphics commands and parameters are sent as character strings. These are sent to a safe area of memory (that is, away from where the operating system and program are) where two

256 kilobit buffers exit. There is a buffer for output to the card, for input from the card, and for errors. A read and write pointer exists for all of these buffers and must be incremented whenever a read or write process is done. Assembly language routines have been written to:

- send character strings to the output buffer,
- read character strings from the input buffer,
- check for errors, and
- manage the buffer pointers.

The communications interface is organized as shown in Figure 3.7.

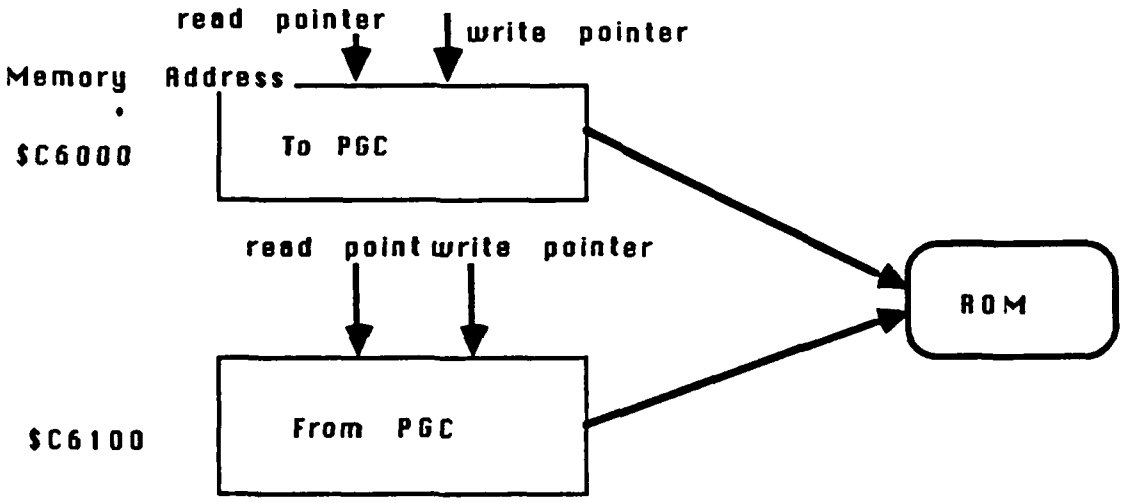


Figure 3.7 Communications to the PGC

To use the assembly language program from FORTRAN, the programmer must say:

call com(hfg command - characater*n)
call infifo(data string read from card -characater*n)

The assembly language program makes communicating with the card simple. For flowcharts and specific details on the assembly language programs, please refer to Appendix B.

3.6 The User Interface

The user-interface, the part of the program that determines how the user and the computer communicate, will make the software useful to other users. The interface is the most difficult to develop and its performance is the hardest to predict. Research at the Media Laboratory at Massachusetts Institute of Technology [Newman and Sproul, 1979][Tufte, 1983] is directed at designing effective user-interfaces. This research project, although not directly concerned with those questions, has developed a user interface for the program consisting of a series of menus. The user's manual, which describes each of the menus in detail, is in Appendix D and should be read by anyone planning to use the program.

3.7 Conclusion

The software was developed to handle many different data sets generated from different methodologies. With quick and precise plotting, more examples can be completed is a shorter period of time. The software will also allow for more complete analysis of the loci with the rotation and translation options in viewing. As more examples are developed and analyzed, the methodologies will be further supported. For example, in SEA, the application of the methodology to different systems and tasks will further demonstrate the general nature of the analysis.

CHAPTER IV

MATHEMATICAL ANALYSIS

4.1 Introduction

This chapter has two parts. First, it describes TACFIRE, an existing Command, Control, and Communication (C^3) system that has been developed by the United States Army for fire support. Second, it presents a mathematical model of the TACFIRE system. Cothier (1984) carried out the effectiveness analysis of this system and assessed its timeliness. The description of the system and the first part of the analysis in this chapter will closely follow the development by Cothier. The second part of the analysis will extend the model of the TACFIRE system to a more realistic situation. The effectiveness of this new model will be assessed in Chapter 6.

4.2 The TACFIRE System

The TACFIRE system is a combination of hardware which operates under a set of standard operating procedures called doctrines and options to which systems effectiveness methodology is applicable. The system can be broken down into three main components: 1) the forward observer (FO), 2) the battalion fire direction center (BN FDC), and 3) the field artillery cannon battery. A number of forward observers and batteries can be connected through the central battery computer , the BN FDC. The forward observer (FO) receives the initial stimulus by detecting an enemy threat. The FO makes estimates of the position and velocity of a threat with position-determining equipment. These estimates are sent to the central computer through a Digital Message Device (DMD) that transmits and receives digital signals. The Battalion Fire Detection Center contains a computer that can receive input and send output digitally by Standard Army Communications , over radio or wire. The Field Artillery Battery contains a Battery Display Unit (BDU) which prints fire data recieved from the central computer. This equipment cannot send data but will send acknowledgement to the central computer that the data was received. In addition to this equipment, voice communicatiion links act in parrallel with the digital links. Although

voice links are slower and more vulnerable in an attack, they are useful if the digital links fail. Moreover, if the TACFIRE system is inoperable, each battery can compute firing data with a field artillery digital automatic computer (FADAC), although this method is slower than TACFIRE.

4.3 The System Model

To apply the system effectiveness methodology, the context, the mission, and the system attributes and primitives must be defined. The geometry of the system in its context is shown in Figure 4.1. which specifies the scenario, and shows the enemy (or threat) is headed on the only road to headquarters. TACFIRE consists of one or two FOs, one BN FDC and two batteries positioned to protect this access. The mission of TACFIRE is to prevent the attack by destroying the threat before it can inflict damage on the headquarters. For simplicity, the only countermeasure the enemy may take is to jam the communication lines. Also, the threat will continue to move towards its goal as if no shots were being fired. These assumptions are not realistic in that they do not reflect the tactics that will be used in such a situation. However, the objective is not to study the effectiveness of TACFIRE per se, but rather to illustrate how the methodology would be applied to an actual system.

The system attributes can be described by two concepts:

Window of Opportunity: Let t^* denote the earliest time at which the threat can be hit and t^{**} the latest time the threat can be hit. To characterize the window of opportunity, the analysis uses the ordered pair $(t^{**}, \Delta t)$ where $\Delta t = t^{**} - t^*$.

OPK: The Overal Probability of Kill is a function of the system characteristics such as hardware and standard operating procedures. It measures the overall capability of the system. The mission requirements can be expressed by this attribute: TACFIRE is required to destroy the threat with a particular probability, which is expressed in terms of an OPK. Because the mission is just concerned with the outcome of the fire support, the window-of-opportunity attributes are not considered.

The system attributes are functions of the system primitives, which represent the building blocks of the model. Reliability is determined by the probability of failure of each node and each link, and is independent of the countermeasures taken by the enemy.

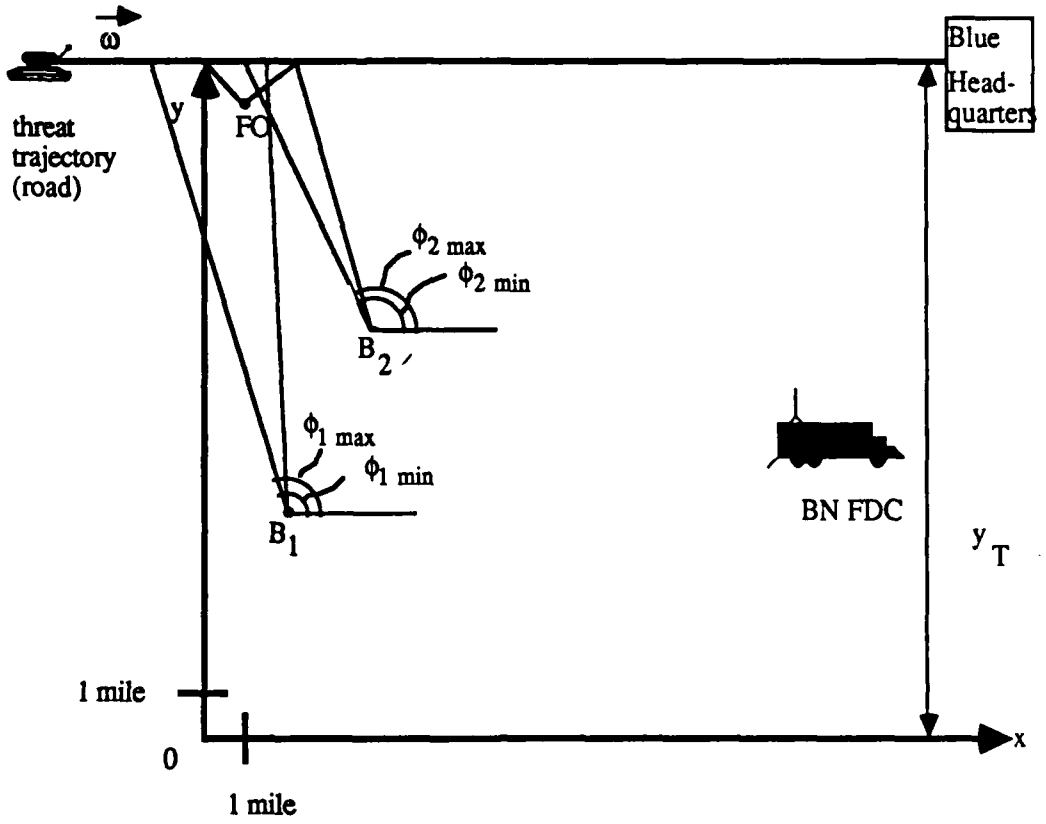


Figure 4.1 Geometry of the System and its Components

Survivability is determined by the probability of failure due to jamming by the enemy. These probabilities are merged into the vector $\underline{p}$, the set of probabilities for each node and link in the system:

$$\underline{p} = (p_1, p_2, \ldots\ldots, p_7)$$

$$p_i \equiv \text{probability that link } i \ (i = 1,\ldots,7) \text{ works}$$

The influence of the scenario on the system is represented by choosing ω, the speed of the threat, as a system primitive. The issue of uncertainty can be represented by the angle β which separates the two sightings of the FO. The only uncertainty comes from the threat estimate by the forward observer. Thus the larger the angle β is, the more accurate the

the speed estimates, but the response time is longer.

In summary, the system attributes are

t^{**} : upper limit of the window of opportunity for the system to respond to the threat,

Δt : width of the window of opportunity, $\Delta t = t^{**} - t^{*}$,

OPK: overall probability of kill,

and the system primitives are

$\underline{p} = (p_1,....,p_7)$: the vector of probabilities that the nodes and the links of the system will operate,

β : the angle between the two observations by the forward observer,

ω : the speed of the threat.

The mission requirements can be expressed at the attribute level in terms of OPK. If λ is the lower bound on the probability of kill, with $0<\lambda<1$, then the mission locus is part of the attribute space (t^{**}, Δt, OPK) defined by OPK $\geq \lambda$.

4.4 Mathematical Derivation

In this section, the equations from which the system's attributes are generated are derived. First, the geometriy is considered for the one-battery case under two seperate doctrines. Then the two-battery case is analyzed with two options for operating the batteries; each option is explored under both doctrines. Lastly, the addition of a second forward observer is considered under both doctrines and options.

As shown in Figure 4.1, the system includes:

- two FOs at (xfo_i, yfo_i) , i = 1,2 with angle of observation constrained between ψ_{min} and ψ_{max}.
- One BN FDC.

• Two batteries, B_i at (xb_i, yb_i) constrained between the two limits $\phi_{i\,min}$ and $\phi_{i\,max}$.

• The threat trajectory path which has the threat moving at a constant speed ω, despite fire from batteries.

The following assumptions about the system are made:

• The first detection time for the first forward observer at angle of observation = ψ_{max} occurs at time t = 0 with $x_t = 0$. Thus, the threat equations of motion are

$$x_t = \omega \cdot t$$
$$y_t = \text{constant} \qquad (4.1)$$

• The threat, the batteries, and the FO's are all considered to be at the same z-coordinate or the same altitude.

• Newtonian equations for the motion of projectiles will be used without condering air resistance.

The system can be modelled as a black box system as shown in Figure 4.2

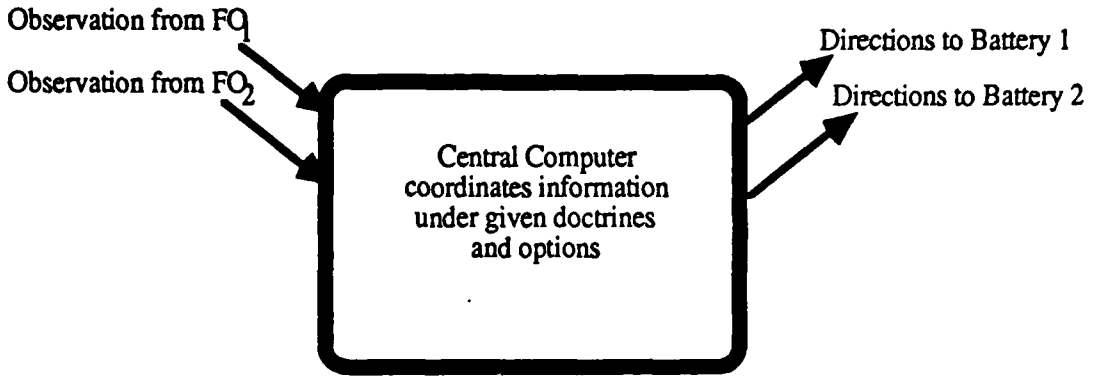


Figure 4.2 Black-box model of the system

Estimates of the speed and position of the threat are received by the central computer in the order they occured in time. The observations are made according to the rules of the particular doctrine being applied. The observation data is coordinated by the computer. From that data, firing data is computed and sent to the batteries under the rules specified in the options. Theorectically, a system can have a number of forward observers and batteries but only two FOs and two batteries will be considered here. Note that the batteries and the forward obsrevers have no direct communication with each other.

4.4.1. Analysis of the Time Profile and the Window of Opportunity

This analysis follows closely Cothier (1984), and thus the presentation here will be succint.

From the time of observation to the time of impact, three distinct events occur as shown in Figure 4.3.

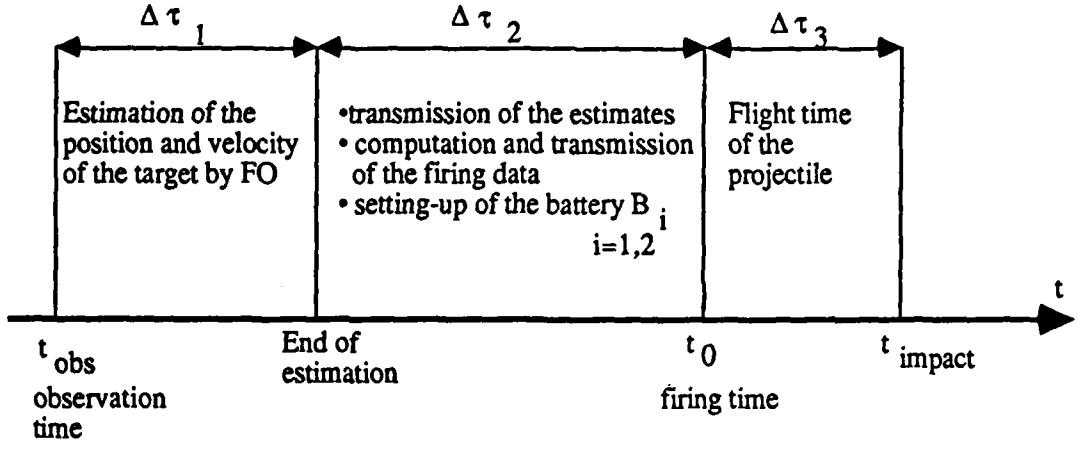


Figure 4.3 Time Profile

$$t_{impact} = t_{obs} + \sum_{i=1}^{3} \Delta\tau_i \qquad (4.2)$$

The geometry of the situation is described in Figure 4.4.

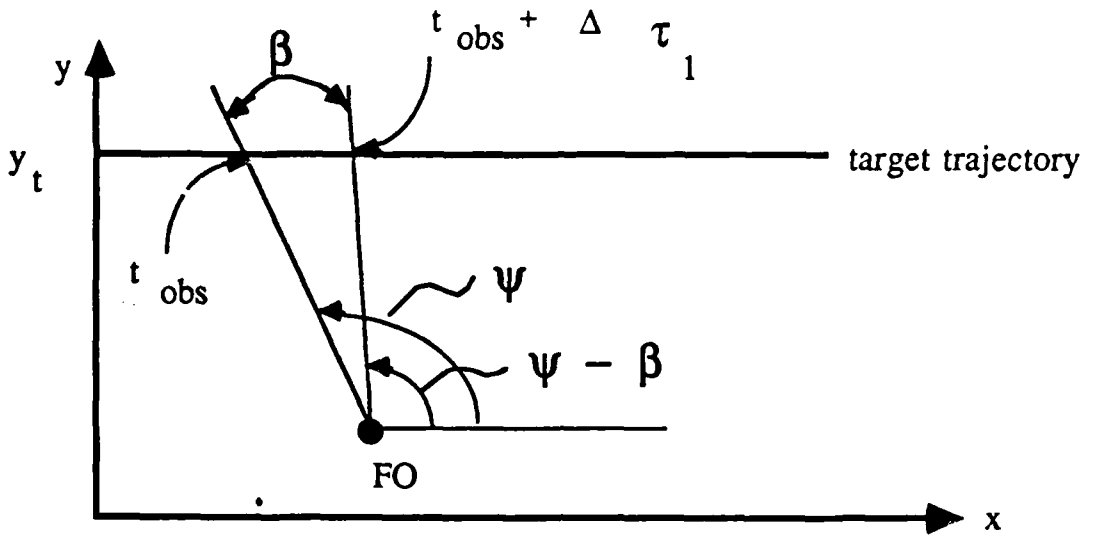


Figure 4.4 Geometry of Observation

From Figure 4.4,

$$\tan \psi = (y_t - y_{FO})/(x_t(t_{obs}) - x_{FO})$$

$$x_t(t_{obs}) = \omega \cdot t_{obs} \qquad (4.3)$$

and thus,

$$\Delta \tau_1 = [(y_t - y_{FO})/\omega] \cdot [(1/\tan(\psi-\beta)) - (1/\tan(\psi))] \qquad (4.4)$$

The time of transmission and setting of the battery is characterized by $\Delta\tau_{2i}$, where i is the path of transmission. Information can move over different media, such as voice, digital links, or radio waves. A path is a possible combination of links over which information can be transmitted. The time of transmission over each path is a constant and the battery

setup time is also a constant. The total time, a constant $\Delta\tau_{2i}$ is derived in Appendix D.

The time of flight of the proctile is $\Delta\tau_3$, which is derived from the Newtonian equations in Appendix D.

$$\Delta\tau_3 = \Delta\tau_3(\phi_i) = 2\sqrt{[(R/g)*\sin\ \alpha]} \qquad (4.5)$$

The constant $\Delta\tau_3$ is 36 seconds for a projectile from battery 1 and 22 second for a projectile from battery 2.

The window of opportunity is defined by the earliest impact time, t*, and the latest impacat time,t **. The earliest time is computed by the earliest observation time , t_{obs} = 0, with the minimum delays.

$$t^* = \Delta\tau_1(\omega,\beta,0) + \Delta\tau_{2min} + \Delta\tau_3 \qquad (4.6)$$

The geometry defines the last shot a battery can fire which is at time t**. This is labeled M_1 in Figure 4.1 when the threat is at position $(x_t(t^{**}),y_t) = (\omega \cdot t^{**},y_t)$. From Figure 4.1,

$$\tan\ \phi_{i\ min} = (y_t - y_{B1})/(\omega \cdot t^{**} - x_{B1}) \qquad (4.7)$$

Let

$$K = x_{B1} + (y_t - y_{B1})/\tan\phi_{i\ min}$$

then

$$t^{**} = K/\omega \qquad (4.8)$$

Therefore, the window of opportunity can be specified by the ordered pair $(t^{**},\Delta t)$ where

$$\Delta t = \begin{cases} t^{**} - t^{*} & \text{if } t^{*} < t^{**} \\ 0 & \text{otherwise} \end{cases} \qquad (4.9)$$

The window of opportunity is shown in Figure 4.5.

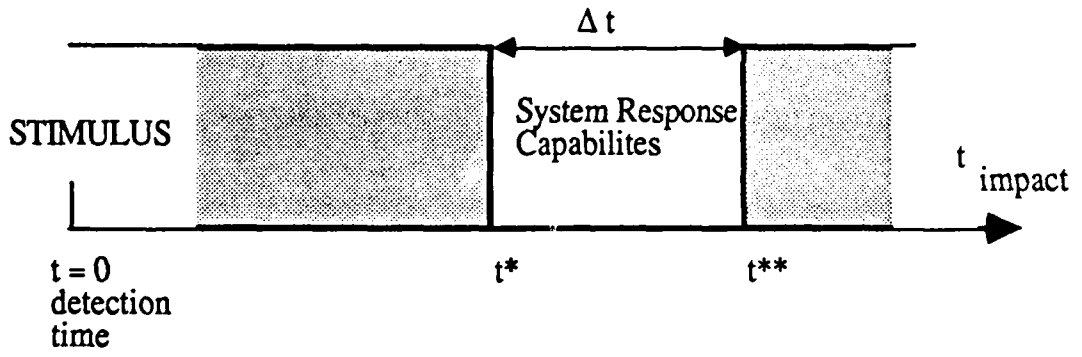


Figure 4.5 The System's Window of Opportunity

4.4.2 Uncertainty Analysis

The only uncertainty in the system is in the estimate of the speed. Let δd denote the error made in the distance estimation, $\delta\beta$ denote the error in the variation of the angle β, and δt denote the error in the time estimate. If $\delta t = 0$ (assume that there is no error in the timing of the estimate), the the relative uncertainty in the estimate of the speed is

$$\rho=\delta\omega/\omega=\delta d/[(y_t - y_{FO})/2)\cdot(1/\sin\ \psi + 1/\sin(\psi-\beta)]+\delta\beta/2\tan(\beta/2) \qquad (4.10)$$

The angle of observation, ψ, is shown is Figure 4.2. The complete derivation of the Equation 4.10 is described in Appendix D. Note that ρ is a decreasing function of β; thus, the larger β is, the more accurate the estimate of the speed.

From the foward observer, the BD FDC receives the estimate of the threat motion . The equations of motion are

$$x_t(t) = x(t_{obs}) + \omega \cdot (t - t_{obs}) \qquad (4.11)$$

Differentiating,

$$\delta x_t = \delta\omega \cdot (t - t_{obs})$$

$$= \omega \cdot \rho(\beta, t_{obs}) \cdot (t - t_{obs}) \qquad (4.12)$$

Thus, from the uncertainty in the estimation of the speed, there is the uncertainy in the position of the threat, which lies in the interval with uniform distribution

$$x_t - \delta x_t \leq \tilde{x}_t \leq x_t + \delta x_t$$

The battery shoots in this uncertainty interval with a kill radius of KR. The probability distribution can be graphed as shown in Figure 4.6

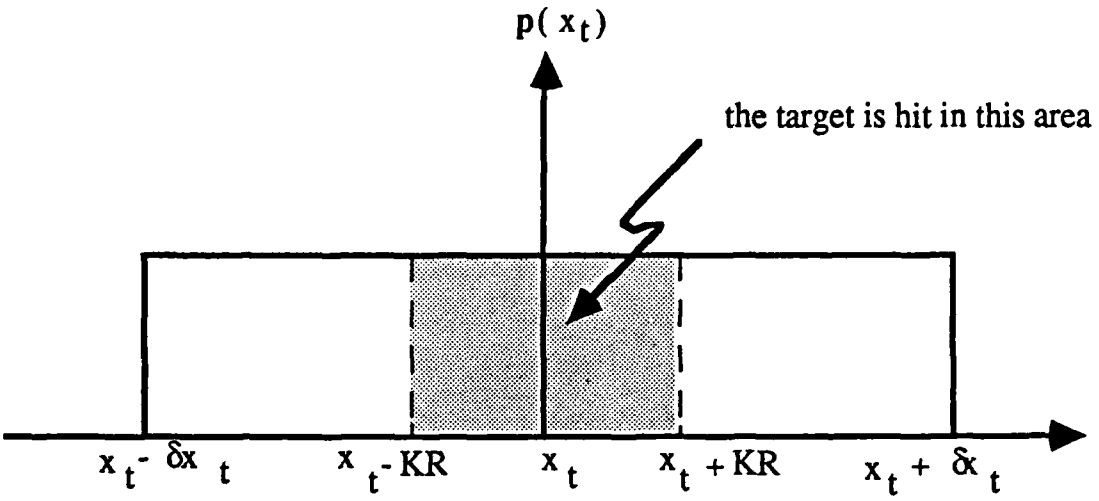


Figure 4.6 Probability Distribution for the Impact Point

Integrating this distribution will give the Single Shot Probability of Kill, SSPK:

$$SSKP(t_{impact}) = \int_{x_t - KR}^{x_t + KR} p(\tilde{x}_t) \cdot d\tilde{x}_t = \frac{2KR}{2\delta x_t} = \frac{KR}{\delta x_t}$$

Note that if KR is greater than or equal to ∂x_t, the integration will yield SSPK = 1 because the complete probability distribution would be integrated. Any part of KR greater than ∂x_t has an SSPK of 0 because the threat is known not to be in that area. Let

$$\xi(\omega,\beta,t_{obs}) = KR / [\omega \cdot \rho(\omega,t_{obs})] \qquad (4.13)$$

then

$$SSPK = \xi(\omega,\beta,t_{obs}) / [t_{impact} - t_{obs}] \qquad (4.14)$$

4.4.3 Analysis of the Network

The links available, as described in the section on TACFIRE are the following:

link #1: Digital Message Device (DMD)
link #2: 1st digital link between a FO and the centeral computer
link #3: BN FDC - central computer
link #4: 2nd digital link between the central computer and a battery
link #5: Voice link between central computer and a battery
link #6: Battaery Dipslay Unit (BDU)
link #7: FADAC/manual equipment

As Cothier demonstrated, only 4 of the 10 possible paths lead to transmission of information from an FO to a battery. The tree network is shown in Figure 4.7, and the network analysis and delays of transmission for each of the four paths are studied in Appendix D. Each path has:

q(i): probability that the path #i is working.

u(i): $\Delta\tau_{2min}(i) + \Delta\tau_3$ which is the minimum delay between the finish time of estimation of the FO and the imptact time.

v(i): the minimum reshooting time; the time needed to recompute new data based on the initial estimates, transmit them, and to set up the battery accordingly.

As shown from Figure 4.7,

$$q(1) = p_1 p_2 p_3 p_4 p_6$$
$$q(2) = p_1 p_2 p_3 p_4 (1 - p_6) p_5 \qquad (4.15)$$
$$q(3) = p_1 p_2 p_3 (1 - p_4) p_5$$
$$q(4) = p_1 p_2 (1 - p_3) p_5 p_7$$

$$u(1) = 92 \text{ seconds}$$

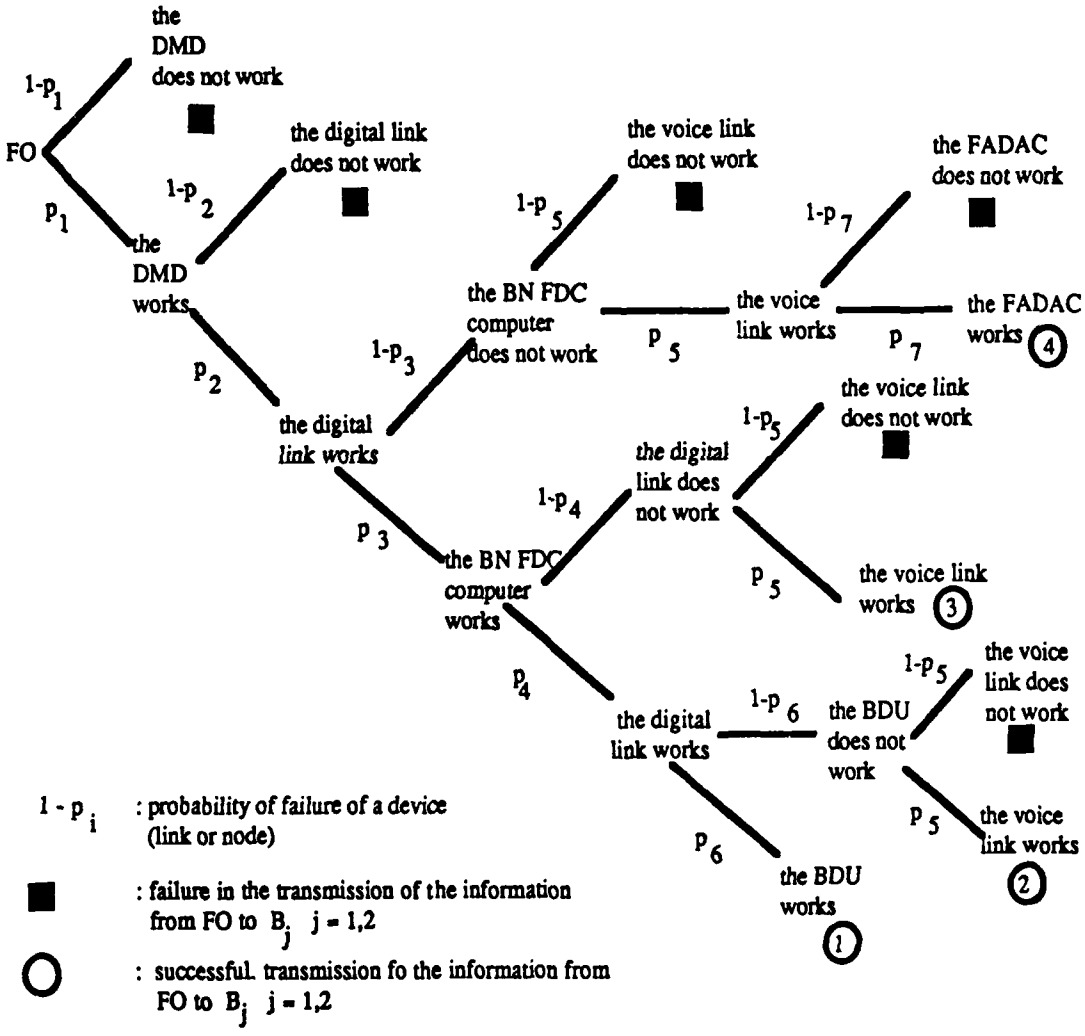
u(2) = u(3) = 109 seconds (4.16)

u(4) = 127 seconds

v(1) = 43 seconds

v(2) = v(3) = 60 seconds

v(4) = 38 seconds



Figuare 4.7 Decision Tree Representing the System Operation Scheme

Again, in Cothier (1984) and in Appendix D, it is shown that the reliabiliaty / survivability function is just the sum of all the probabilities of the 4 paths that successfully transmit.

$$RS(p) = \sum_{i=1}^{4} q(i)$$

The probabilites are all assumed to be equal to p, although in reality this is probably not the case; digital links cannot be jammed while voice links can. The assumption is made here to reduce the dimensionality of the problem for this discussion. The software can accept distinct values for each probability. However, if $p_i = p$ for i=1,...,7 then Equation (4.15) reduces to:

$$\begin{aligned} q(1) &= p^5 \\ q(2) &= p^5(1-p) \\ q(3) &= p^4(1-p) \\ q(4) &= p^4(1-p) \end{aligned} \qquad (4.18)$$

and

$$RS(p) = \sum_{i=1}^{4} q(i) = p^4(2 - p^2) \qquad (4.19)$$

4.5 The Doctrines and the Options

The doctrines and the options are rules which specify how the time is managed in the window of opportunity. Doctines regulate when observations and shots occur and options regulate how the batteries fire in the two battery case. The rules are those stated by Cothier (1984). The two battery case under doctrine 2 and the case of two forward obsersers will be considered in the next section.

The "LOOK-SHOOT-SHOOT-SHOOT..." Doctrine

Definition: the observer initially makes estimates of the speed and position of the threat,

and then the battery keeps on shooting at the threat, recomputing each new firing data on the basis of these initial estimates.

The "LOOK-SHOOT-LOOK-SHOOT..." Doctrine

Definition: After each shot, if the threat is neither destroyed nor incapacitated, the observer makes new estimates of its speed and position, new firing data are computed on the basis of these updated estimates, the battery shoots according to these new firing data, and so on until the upper limit of the window of opportunity is reached.

The equations for each docrine are as follows:

Doctrine 1: $t_{obs} = 0$ and the reshooting time is v(i) for path i = 1,..,4. After the window of opportunity is computed, the number of shots n_i is

$$n_i = 1 + \text{int}[\Delta t_i / v(i)] \qquad (4.20)$$

and the impact time is then

$$t_{impact}(n) = t_i^* + (n-1) \cdot v(i) \quad \text{for } n = 1,....,n_i \qquad (4.21)$$

From equation (4.14),

$$SSPK(t_{impact}(n)) = \xi(\omega,\beta,t_{obs}) / [t_{impact}(n) - t_{obs}(n)]$$

but since $t_{obs}(n) = 0$ for any shot n, then because

$$t^*_i = \Delta\tau_1(\omega,\beta,0) + u(i) \qquad (4.22)$$

$$SSPK(t_{impact}(n)) = \xi(\omega,\beta,t_{obs}) / [\Delta\tau_1(\omega,\beta,0) + u(i) + (n-1) \cdot v(i)$$

$$\text{for } n = 1,....,n_i \qquad (4.23)$$

From Barlow and Proschan (1974), the Overall Probability of Kill is then

$$OPK_i = \coprod_{n=1}^{n_i} SSPK_i\,(t_{impact}(n)) \qquad (4.24)$$

where

$$\coprod_i = 1 - \prod_i (1 - x_i)$$

Substituting in equation (4.23),

$$OPK_i^{(B2)} = \coprod_{n=0}^{m_i - 1} \frac{\xi\,(\omega,\beta,t_{obs})}{\frac{K_2'}{\omega} + n \bullet v(i) - t_{obs}} \qquad i = 1,...,4 \qquad (4.25)$$

Doctine 2: LOOK-SHOOT-LOOK-SHOOT

In this case, the $t_{obs}(n)$ is the last impact time, $t_{impact}(n-1)$. Thus

$$t_{obs}(1) = 0$$

$$t_{obs}(n) = t_{impact}(n-1) \text{ for } n \geq 2$$

$$t_{impact}(n) = t_{obs}(n) + \Delta\tau_1(\omega,\beta,t_{obs}(n)) + \Delta\tau_{2min} + \Delta\tau_3 \text{ for } n \geq 1$$

$$t_{impact}(n) = t_{impact}(n-1) + \Delta\tau_1(\omega,\beta,t_{impact}(n-1)) + u(i) \text{ for } n \geq 2$$

and the number of shots is found by solving the inequality

$$t_{impact}\,(n_i) \leq t^{**} \leq t_{impact}\,(n_{i+1})$$

From equation (4.14) and (4.22),

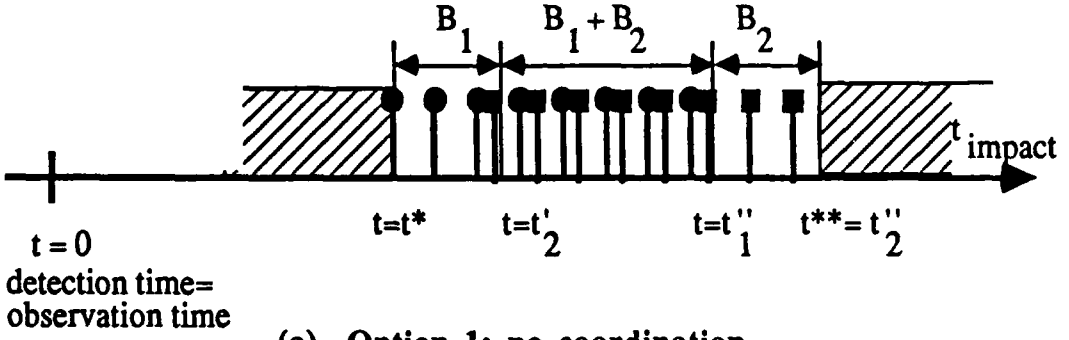
$$SSPK(t_{impact}(n)) = \xi(\omega,\beta,t_{impact}(n-1)) \,/\, [\,\Delta\tau_1(\omega,\beta,t_{impact}(n-1)) + u(i)] \quad \text{for } n = 1,....,n_i \qquad (4.26)$$

From this, and equation (4.24),

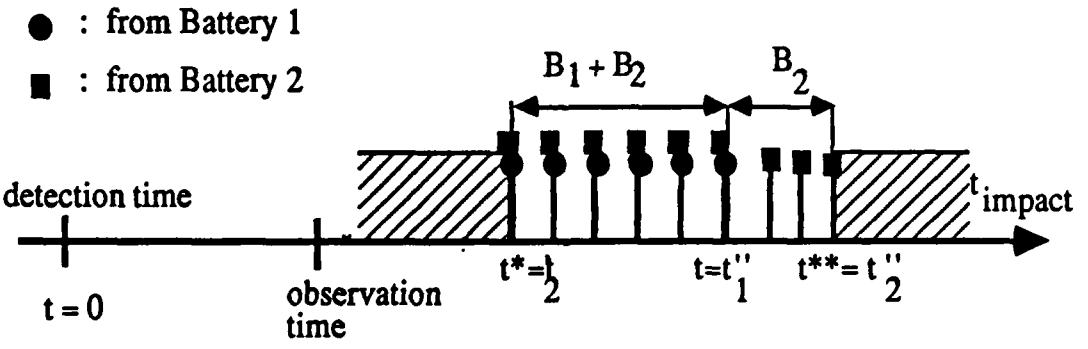
$$OPK_i = \prod_{n=0}^{n_i - 1} \frac{\xi\,(\omega,\beta,t_{impact}(n))}{\Delta\,\tau(\omega,\beta,t_{impact}(n)) + u(i)} \qquad i = 1,...,4 \qquad (4.27)$$

Option 1: The two batteries shoot at the threat independently, each one using the maximum of its own window of opportunity. There is no coordination between the two batteries.

Option 2: Battery B_1 starts firing only when the threat enters the area covered by both batteries. Fire will wait until both B_1 and B_2 can hit the threat at the same time. The global window of opportunity of the system is thus reduced to that of battaery B_2. The window of opportunity is shown in Figure 4.8.



(a) Option 1: no coordination



(b) Option 2: coordination

Figure 4.8 The Window of Opportunity for Both Options

The equations of the options are as follows:
From Figure 4.1,

$$K_2' = x_{B2} + (y_t - y_{B2}) / \tan \varphi_{2\ max}$$

$$K_2'' = x_{B1} + (y_t - y_{B2}) / \tan \varphi_{2\ min}$$

In Appendix D, the following results are derived:

Option 1: no coordination

$$t^{**} = K_2'' / \omega$$

$$\Delta t = \sum_{i=1}^{4} q(i) \cdot OKP_i$$

where $OPK_i = 1 - (1-OPK_i^{(B1)})(1-OPK_i^{(B2)})$ (4.28)

$$OPK_i^{(B1)} = \prod_{n=0}^{n_i - 1} \frac{\xi(\omega,\beta,0)}{\Delta\tau(\omega,\beta,0) + u(i) + n \cdot v(i)} \quad i = 1,...,4 \qquad (4.29)$$

$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega,\beta,0)}{\frac{K_2'}{\omega} + n \cdot v(i)} \quad i = 1,...,4 \qquad (4.30)$$

$$n_i = 1 + \text{int}(\Delta t_i / v(i)) \qquad m_i = 1 + \text{int}[(K_2'' - K_2') / (\omega \cdot v(i))]$$

Option 2: coordination

$$t^{**} = K_2'' / \omega$$

$$OPK = \sum_{i=1}^{4} q(i) \cdot OPK_i \qquad (4.31)$$

where $OPK_i = 1 - (1-OPK_i^{(B1)})(1-OPK_i^{(B2)})$

$$OPK_i^{(B1)} = \prod_{n=0}^{\eta_i - 1} \frac{\xi(\omega,\beta,0)}{\frac{K'_2}{\omega} + n \cdot v(i)} \qquad i = 1,...,4 \qquad (4.32)$$

$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega,\beta,0)}{\frac{K'_2}{\omega} + n \cdot v(i)} \qquad i = 1,...,4 \qquad (4.33)$$

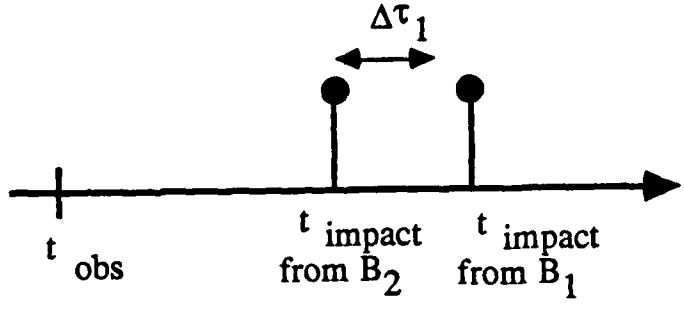
The values of t_{obs} and η_i are derived in Appendix D.

4.6 Introduction of Second Forward Observer

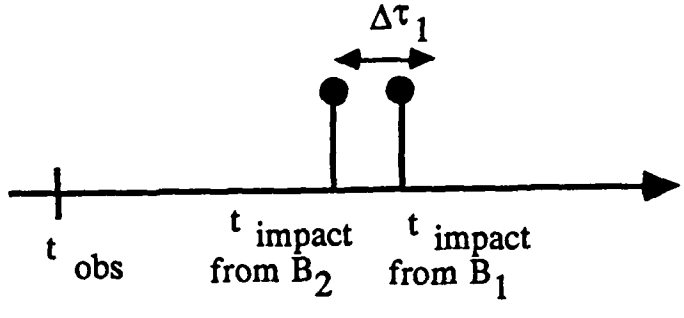
The introduction of a second forward observer makes the model of the TACFIRE system more realistic. The second forward observer will only be considered under Doctrine 2, L.S.L.S., because analysis of the second forward observer in the Doctrine 1, L.S.S.S., is trivial; the only observation used is the first one at t = 0 and then the battery shoots from the information in that estimate. Therefore, this section first presents the derivation of the equations for the two battery case under Doctrine 2. Secondly, the case of two FOs with one battery under option 2 is analyzed and a new primitive and attribute are introduced. Finally, the two-battery case with two FO's is presented.

Doctrine 2, L.S.L.S., Option 1

The rules are the same as in Doctrine 1, Option 1. The FO makes an observation and battery one and battery two shoot independently from this estimate. As shown in Figure 4.1, battery 2 is closer to the threat trajectory than battery 1; thus, the time of flight is less (assuming the projectile lands on the path), and the impact times will not be the same. The next observation will be at the next possible impact time. There are two possible times that the next observation can take place, as shown in Figure 4.7



(a) Observation time less than the time betwen impacts



(b)Observation time greater thanthe time between impacts

Figure 4.8 Criteria to choose next observation time

The next observation occurs at the impact time from the previous observation. If an impact occurs while the forward observer is making an observation, then the next observation can not be made at this particular impact time. This is shown in Figure 4.8 (b).
If the impact occurs after the observation, then an observation is possible. This is shown in Figure 4.8 (a). Thus, two observations may occur very close in time to each other.

Doctrine 2, Option 2

Again, the rules are applied in the same way. The batteries shoot so that their impact times are the same. Because the flight time from battery 2 is less than that of battery 1, battery 2 may have to wait to send the projectile.

The impact times are calculated from the inequality from Doctrine 2 and the window of opportunity is the same as in the two-battery, doctrine 1 case (the window of opportunity is determined by the geometry and the option, not the doctrine).

The equations for each case are given below.
Doctrine 2, Option 1:

Let $t_{impact\ 1}$ be the impact from battery 1 and $t_{imapct\ 2}$ be the impact from battery 2. After each impact time is computed, the following check must be made:

$$\text{if } abs(t_{impact\ 1}^{(n-1)} - t_{impact\ 2}^{(n-1)}) \leq \Delta\tau_1 \text{ then}$$

$$\text{if } t_{impact\ 1}^{(n-1)} \leq t_{impact\ 2}^{(n-1)} \text{ then } tobs(n) = t_{impact\ 1}^{(n-1)}$$

$$\text{else } t_{obs}(n) = t_{imapct\ 2}^{(n-1)}$$

Let n_i denote the number of shots from battery 1 and m_i the number of shots from battery 2. Then

$$OPK_i = \prod_{n=0}^{n_i - 1} \frac{\xi(\omega,\beta,t_{obs}(n))}{t_{impact\ 1}(n) - t_{obs}(n)} \qquad i = 1,...,4 \qquad (4.34)$$

$$OPK_i = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega,\beta,t_{obs}(n))}{t_{impact\ 2}(n) - t_{obs}(n)} \qquad i = 1,...,4 \qquad (4.35)$$

As before

$$OPK_i = 1 - (1-OPK_i^{(B1)})(1-OPK_i^{(B2)}).$$

Doctrine 2, Option 2

The number of shots is determined by battery 2 and the window of opportunity is determined by where battery 2 can shoot. Let μ_i be the number of shots from battery 2 and m_i the number of shots from battery 1. Consider the doctrine 2 with just battery 2. The first observation, $t_{obs} \neq 0$, is determined in Appendix D from the geomerty of battery 2 as is the window of opportunity. By definition of option 2, every time fire from battery 2 impacts the threat trajectory, fire from battery 1 impacts at the same time. The number of shots from battery 2, μ_i, is determined by the inequality from Doctrine 2. The number of shots from battery 1, m_i, is determined by how many shots can be fired by this battery constrained by the geometry of the window of opportunity. The BN FDC keeps track of the observation and impact times and the equations are the same as above.

The Second Forward Observer

A detailed look of the geometry of a second forward observer is shown in Figure 4.10.

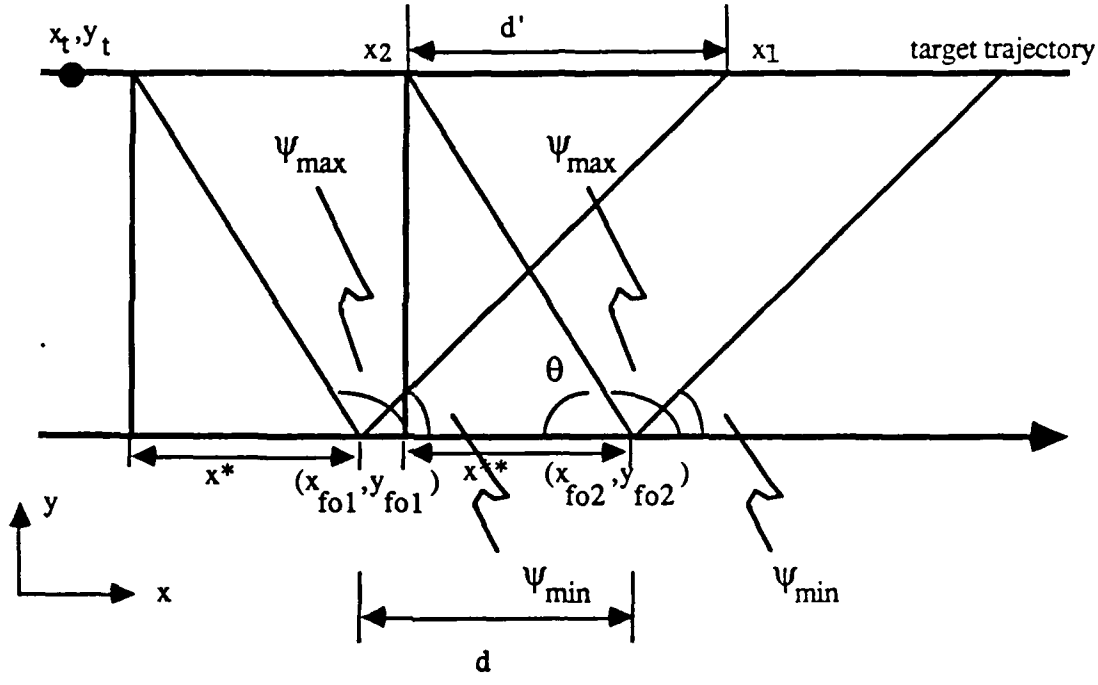


Figure 4.10 Geometry of Two Forward Observers

The y coordinates of the forward observers are the same, as are the angles which constrain the observation. The only thing that changes is the distance between them. This distance is a system primitive.

$$d = x_{fo2} - x_{fo1} \quad \text{with} \quad x_{fo2} \geq x_{fo1} \tag{4.36}$$

$$\tan \psi_{min} = (y_t - y_{fo1}) / x^* \qquad x^* = (y_t - y_{fo1}) / \tan \psi_{min}$$

$$x_1 = x_{fo1} + x^* = x_{fo1} + (y_t - y_{fo1}) / \tan \psi_{min} \tag{4.37}$$

Let $\theta = 180 - \psi_{min}$. Then,

$$\tan \theta = (y_t - y_{fo2}) / x^{**} \qquad x^{**} = (y_t - y_{fo2}) / \tan (180 - \psi_{min})$$

$$x_2 = x_{fo2} + x^{**} = x_{fo2} + (y_t - y_{fo2}) / \tan (180 - \psi_{min})$$

Let d' be the distance between the maximum point that FO_1 can observe on the threat trajectory and the minimum point FO_2 can observe on the threat trajectory. As shown in Figure 4.10:

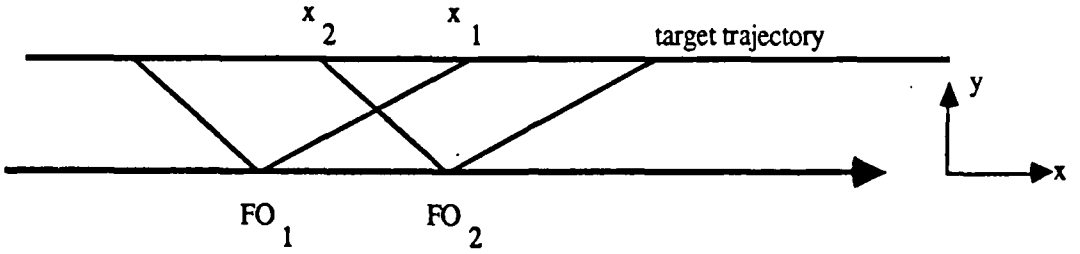
$$d' = x_2 - x_1 \qquad (4.38)$$

if d' < 0 then the range of observation of the 2 FO's intersects

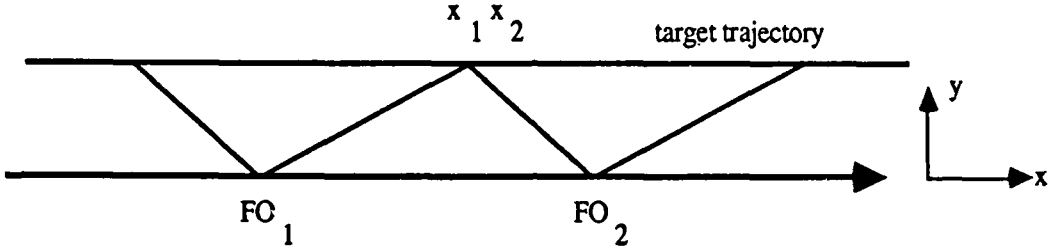
if d' = 0 then the ranges of observation of the 2 FO's are *touching*

if d' > 0 then there is a "blind spot" on the threat trajectory in thewindow of opportunity where no observations take place. Doctrine 1, L.S.S.S., is applied for this area, with the last observation serving as the estimate for all the shots until the next observation can occur.

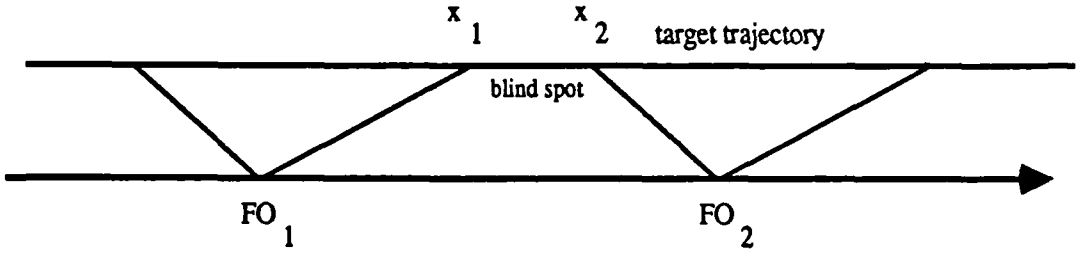
These three cases are shown in Figure 4.11 (a), (b), and (c). This distnance, d',a function of d, is the distance between the forward observer. The distance d and d' will varied and used to determine if a second forward observer is worth having.



(a) Range of observation touches



(b) Range of observation touches



(c) Blind Spot

Figure 4.11 Forward Observation Cases

The equations for the second forward observer are the same as for the one forward observer case under Doctrine 2 with two batteries. Each option is applied with each observer whose observations are independent from the other's. Once the SSPKs are computed, the OPK can be calculated as follows:

Let $SSPK_{11i}$ be the SSPK for path i with FO_1 with shots from battery 1

$SSPK_{12i}$ be the SSPK for path i with FO_1 with shots from battery 2

$SSPK_{21i}$ be the SSPK for path i with FO_2 with shots from battery 1

$SSPK_{22i}$ be the SSPK for path i with FO_2 with shots from battery 2

Then,

$$OKP_i = 1 - \left[\prod_{n=0}^{n_i - 1} (1 - SSPK_{11i}) \prod_{n=0}^{m_i} (1 - SSPK_{21i}) \right] \quad (4.23)$$

where n_i is the total number of shots fired from battery 1, and m_i is the total number of shots fired from battery 2.

The blind spot with two forward observers occurs when the areas of observation do not intersect. Doctrine 1, L.S.S.S, is applied with the last observation serving as the estimate for which the shots are fired. Doctrine 2, L.S.L.S., resumes when FO_2 makes his first observation. This case is discussed in Chapter 6.

A special case occurs when the forward observers are located on the same spot, that is d=0 and $x_{fo1} = x_{FO2}$. In probabilistic terms, each observer obtains a sample of the same event. Remember that the FO's are exactly the same except for the uncertainties in their estimates. The information in the estimates is fused into one estimate for the batteries. This is done by averaging the uncertainties, which are δd, the uncertainty in the distance measurement, and $\delta\beta$, the uncertainty in the angle of estimation. Once these are averaged, the new uncertainties are used to compute the firing data as if there was one observation. If the uncertainties are the same, the OPK's will be the same as just FO_1. If the second foward observer has higher uncertainty that the first, then the overal uncertainty in the estimate will increase and the OPK's will decrease; if the second forward observer has a lower uncertainty than the first, then athe overall uncertainty in the estimate will decrease and the OPK's will increase.

If the forward observers are very close to each other, then the estimates will call for the batteries to shoot in succession. If, however, a message is sent to the battery to shoot

while it is reloading, there must be a wait period before the battery can act on the estimate. This only occurs for certain geometries and is discussed in Chapter 5.

4.7 Summary of the System and the Mission

The system attributes and primitives are summerized below:

- System primites (independent variables)

  ω: speed of the threat

  β: variation in FO's observation angle

  p: probability that any of the 7 links work

- System attributes

  t^{**} : upper limit of the system window of opportunity

  Δt : width of the system window of opportunity

  OPK: overall probabilty of kill of the system

In Figure 4.13, the map of the primitives into the attributes is shown. Note that the window of opportunity and the distance between the two FOs are mapped into the OPK from the doctrines.

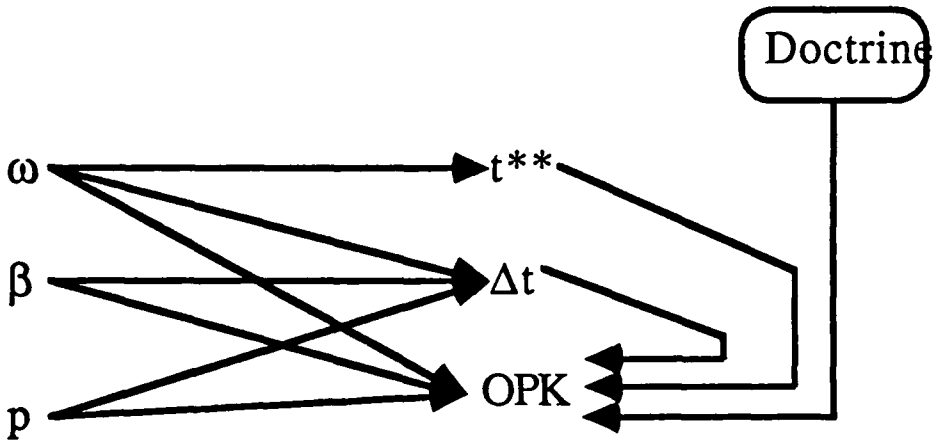


Figure 4.13 Map of System Primitives into System Attrributes

OKP and Δt refer to the expected values of the overall probability of kill and the width of the window of opportunity for each path. These expected values are expressed as:

$$E(\Delta t) = \sum_{i=1}^{4} q(i) \cdot \Delta t_i \qquad (4.24)$$

$$E(OPK) = \sum_{i=1}^{4} q(i) \cdot OPK_i$$

4.8 System and Mission Locus

The system locus is generated by varying the primitives over their allowable ranges and generating a surface in attribute space. Only three attributes can be viewed at a time. At each unique primitive combination (ω,β,p,d), an attribute point is generated (t**,Δt, OPK, d'). As stated earlier, the mission is defined in terms of OPK and thus, the mission locus is just the half space in the positive orthant of the attribute space (or everything above the horizontal plane OPK = λ).

The loci from each of the cases discussed above, generated by the graphics software, will be discussed in the next chapter. Comparisons with examples from Cothier (1984) will be drawn and the effect of a second forward observer will be assessed.

CHAPTER V

ANALYSIS OF THE DOCTRINES AND OPTIONS USING THE GRAPHICS SYSTEM: ONE OBSERVER CASE

5.1 Introduction

In this chapter, the results of the mathematical analysis are presented. The system considered is the one described in Chapter 4 with one forward observer. The following cases are considered:

- Doctrine 1 (L.S.S.S.), 1 battery
- Doctrine 2 (L.S.L.S.), 1 battery
- Doctrine 1, Option 1 (2 batteries, uncoordinated)
- Doctrine 1, Option 2 (2 batteries, coordinated)
- Doctrine 2, Option 1 (2 batteries, uncoordinated)
- Doctrine 2, Option 2 (2 batteries, coordinated)

The first four cases listed were previously analyzed in Cothier (1984), but all the results has been recomputed with a TURBO Pascal program on the IBM-PC/AT (see Appendix A for specifications). The replication of this work has served as a benchmark for developing the graphics system and the new software which generates the data. Cothier's analysis has been enhanced with the use of the graphic system which allows for different views of the data. All the loci figures in this chapter have been generated by the graphics system described in Chapter 3. Thus, this chapter will not only present the results, but also demonstrate how the graphics system can be used as a tool for viewing loci.

The numerical values for the system are taken directly from Cothier (1984) and results here will be compared to earlier work. They are as follows:

Geometric parameters:

x_{B1} = 3 miles (4,827 m) $\phi_{1\ min} = 96°$

y_{B1} = 5 miles (8,045 m) $\phi_{1\ max} = 110°$

x_{B2} = 4 miles (6,436 m) $\phi_{2\ min} = 105°$

y_{B2} = 9 miles (14,481 m) $\phi_{2\ max} = 115°$

x_{FO} = 1 mile (1,609 m) $\psi_{min} = 15°$

y_{FO} = 14 miles (22,526 m) $\psi_{max} = 135°$

y_T = 15 miles (24,135 m)

System parameters:

δd = 16' 5" (5 m)

$\delta\beta = 0.2°$

KR = 65' 7" (20 m)

Range of Variation for System Primitives:

10 mph $\leq \omega \leq$ 20 mph (4.469 m /sec $\leq \omega \leq$ 8.929 m / sec)

$5° \leq \beta \leq 15°$

$0.85 \leq p \leq 0.95$

Mission Requirements:

$1 \geq OPK \geq \lambda = 0.75$

5.2 Remarks on Analysis

Before the results are discussed, it is necessary to examine certains details of the anlaysis. These details will help in understanding the loci presented later.

5.2.1 Measure of Effectiveness: Mission and System Locus

The mission requirement, translated in terms of the third attribute OPK, is the minimum probability of incapacitating the target. If that level is denoted by λ, with $0 < \lambda \leq 1$, then the mission locus is the slab defined by horizontal plane between $\lambda \leq OPK \leq 1$ for $\Delta t \geq 0$ and $t^{**} \geq 0$ in the attribute space $(\Delta t, OPK, t^{**})$. Figure 5.1 shows the mission locus in the attribute space.

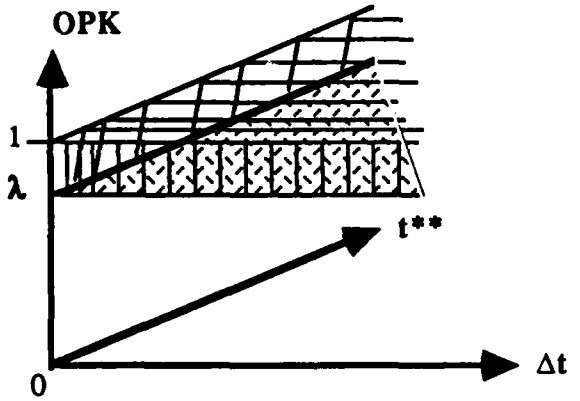


Figure 5.1 The Mission Locus

To obtain a measure of effectiveness, Cothier (1984) chose $\wp(L_s)$ to be the volume of the system locus, and $\wp(L_s \cap L_m)$ to be the intersection of the system and mission loci so that the measure of effectiveness E is the ratio of the two volumes:

$$E = \frac{\wp(L_S \cap L_m)}{\wp(L_S)} \qquad (5.1)$$

The system locus is derived by varying the primitives over their allowable ranges. In Figure 5.2, a line of the locus is generated by varying β, with ω and p fixed. In the graphics program, this line of the locus is plotted by varying the index **j**, with the indices **i**, **k**, and **l** fixed. Because ω is fixed, t** (the third attribute) is also fixed, and the line is generated on the (Δt, OPK) plane.

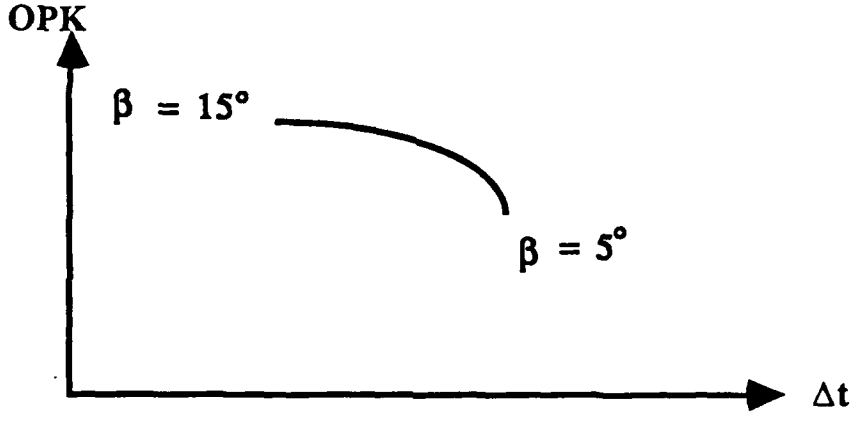


Figure 5.2 Variation of β with ω and p fixed

If p also varies over the allowable ranges, with ω fixed, a family of curves will be generated, as shown in Figure 5.3; this continuous set of curves sweeps out a surface. This is plotted in the graphics program by varying the index **k**.

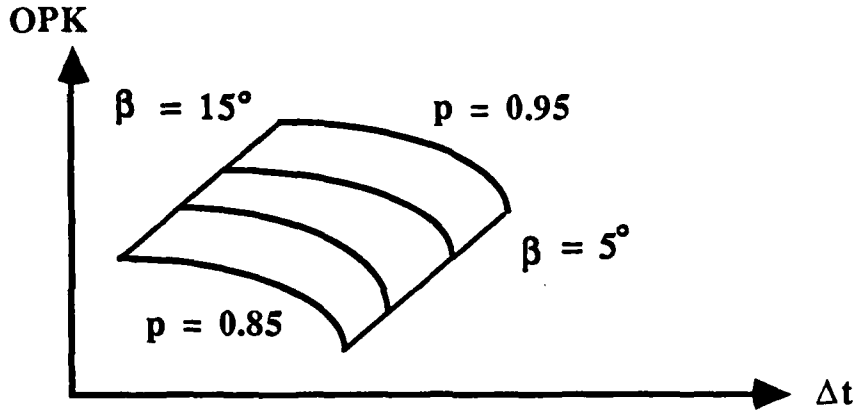


Figure 5.3 Variation in p and β, with ω fixed

Finally, as the primitive ω varies, a family of surfaces or cross-sections of the locus are generated in the attribute space (Δt, OPK, t**). This family of cross-sections sweeps out the volume of the locus. Let S(ω) denote the surface or cross-section associated with a particular ω. The volume is :

$$\mathcal{V}(L_S) = \int_{\omega_{min}}^{\omega_{max}} S(\omega) \cdot d\omega \qquad (5.2)$$

As suggested above and described in Chapter 3, the locus is plotted in the same style as it is computed. Figure 5.4 (a) shows the data points as they exist in the attribute space. These points are connected by different lines when the indices are varied. The indices correspond to the primitives which are varied to generate the loci data. Figure 5.4(b) shows lines which connect the points as p varies (corresponding to index k). Figure 5.4(c) shows the locus as a family of cross-section and the second set of lines which connect the points as β varies (corresponding to the index j). Finally, Figure 5.4(d) shows the complete locus and the lines which connect the points a ω varies (corresponding to the index i). The values over which the primitives and the indices range is indicated in Figures 5.4(a)-(d).

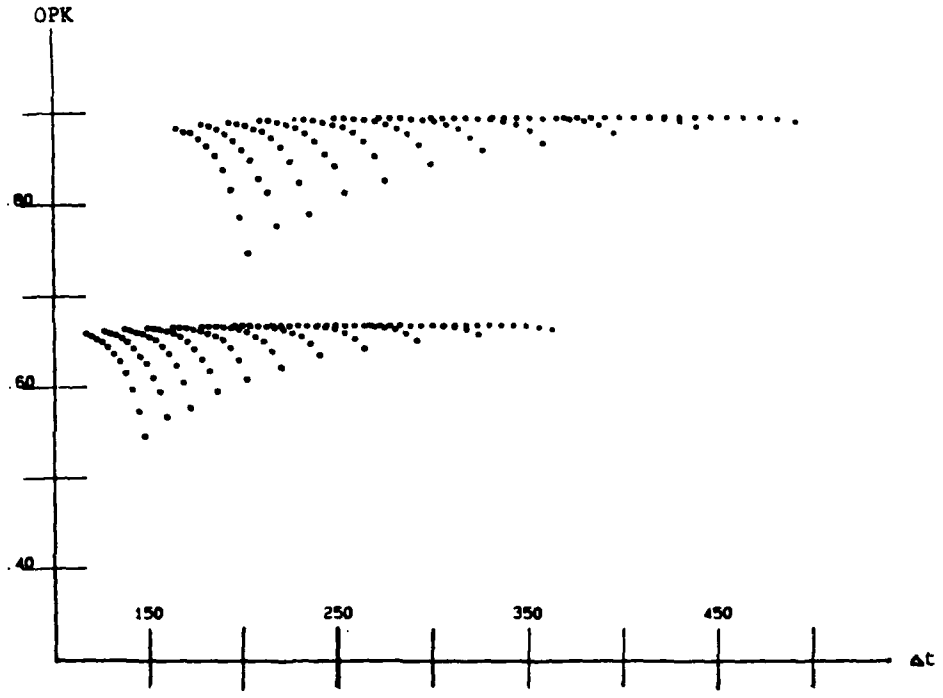


Figure 5.4 (a) Data Points of the System Locus

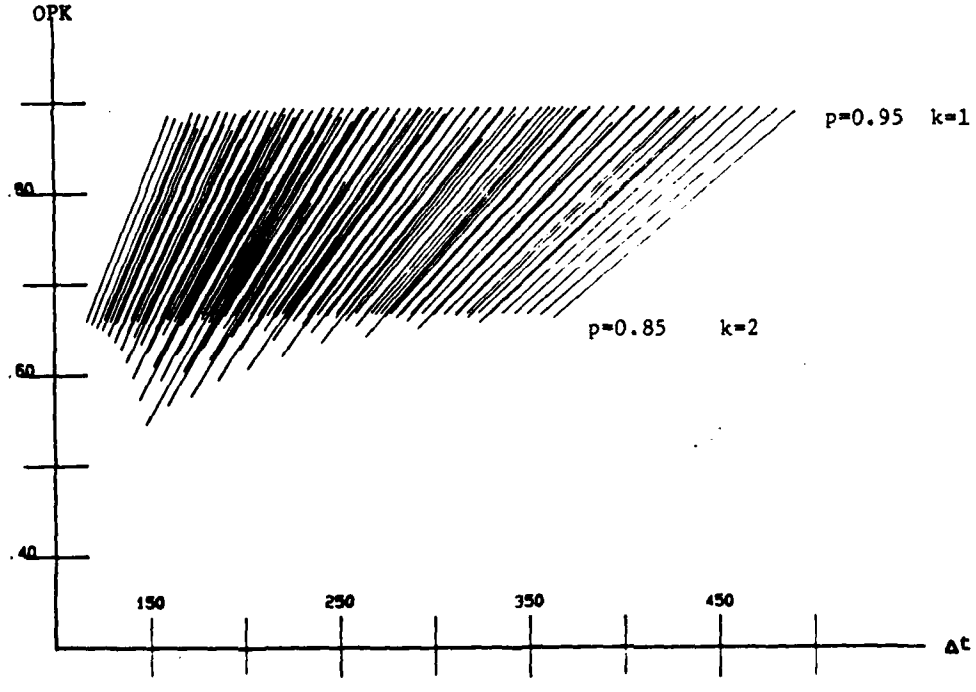


Figure 5.4 (b) Data Points with the Same w and B Connected by Variation in p

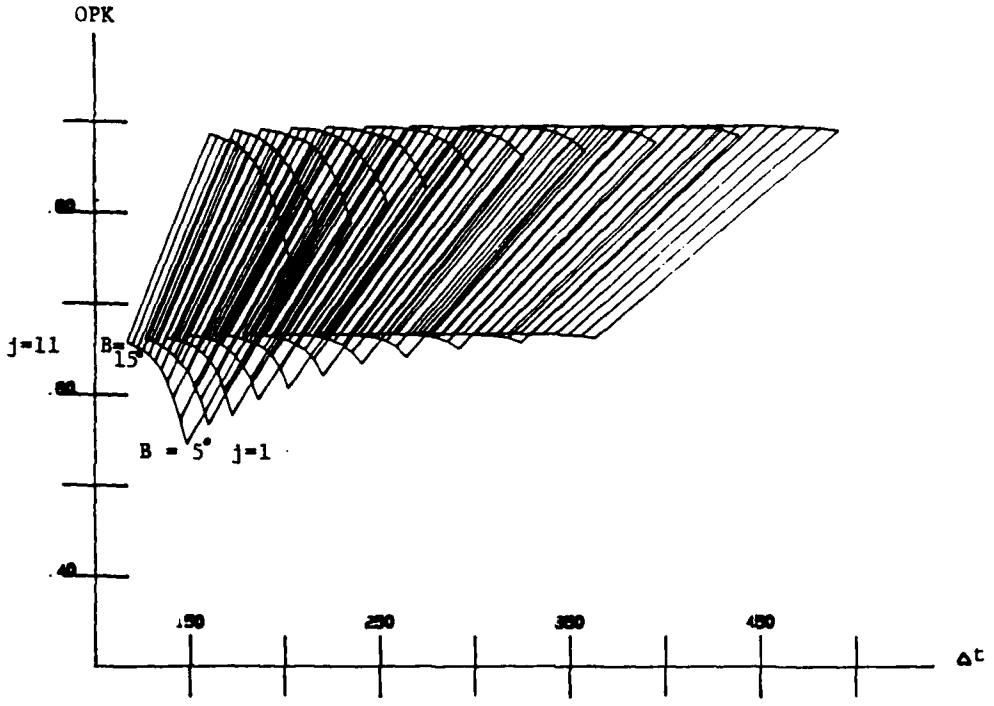


Figure 5.4 (c) Data points connected with varying B - cross sections

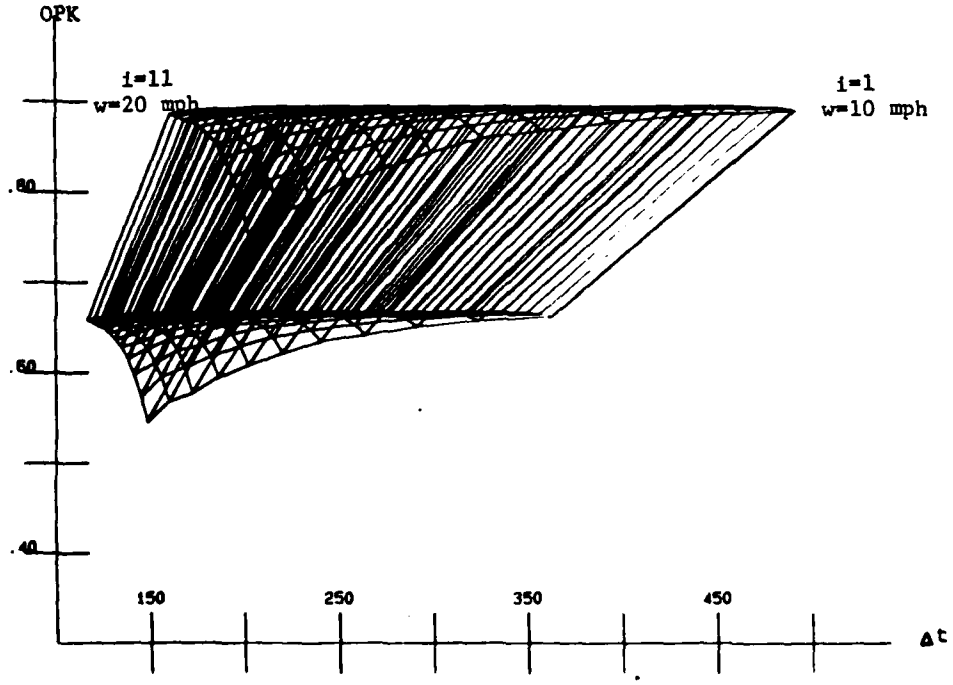


Figure 5.4 (d) Data points connected by varying w - full locus

5.2.2. Upper Limit in Loci Values

The OPK of any locus does not go beyond the value of 0.8939 which is precisely the value of the reliability / survivability function at p = 0.95. This is the highest probability that the link is working. Thus, for each path, the system can be improved to yield OPKs approximately equal to one; but unless the probabilites associated with the network improve, the overall performance will be bounded by 0.8939. This is expressed mathematically as:

$$OPK = \sum_{i=1}^{4} q(i) \cdot OPK_i \approx \sum_{i=1}^{4} q(i) = RS(p) = p^4(1 - p^2) \qquad (5.3)$$

where **RS (p)** is the reliability/survivability function for the system, and **q(i)** is the probability that path #i is working. Thus,

$$OPK_{max} = RS(p_{max}) = (0.95)^4 \cdot (2 - (0.95)^2) = 0.8939$$

In summary, the system cannot be further improved unless the probabilities associated with the network are also improved.

5.2.3 Computational Errors

If one looks closely at Figure 5.4(d), a small discontinuity in the locus can be observed on the lower and upper left edge. This locus, the system locus for Doctrine 1, was generated by Cothier(1984) with smooth lines. Figure 5.5 shows the same locus with more cross-sections generated and the discontinuities still exist. Taking more samples will not change this phenomenon. This occurs from using the *round* operation when computing the number of shots: a real number must be converted into an integer number [Chapra and Canale, 1985]. Then *round* operation mathematically is expressed by the following rule:

Num, a real number, is rounded to the nearest integer as follows:
if Num ≥ 0 then Round(Num) = Trunc(Num + 0.5)
if Num < 0 then Round(Num) = Trunc(Num - 0.5).

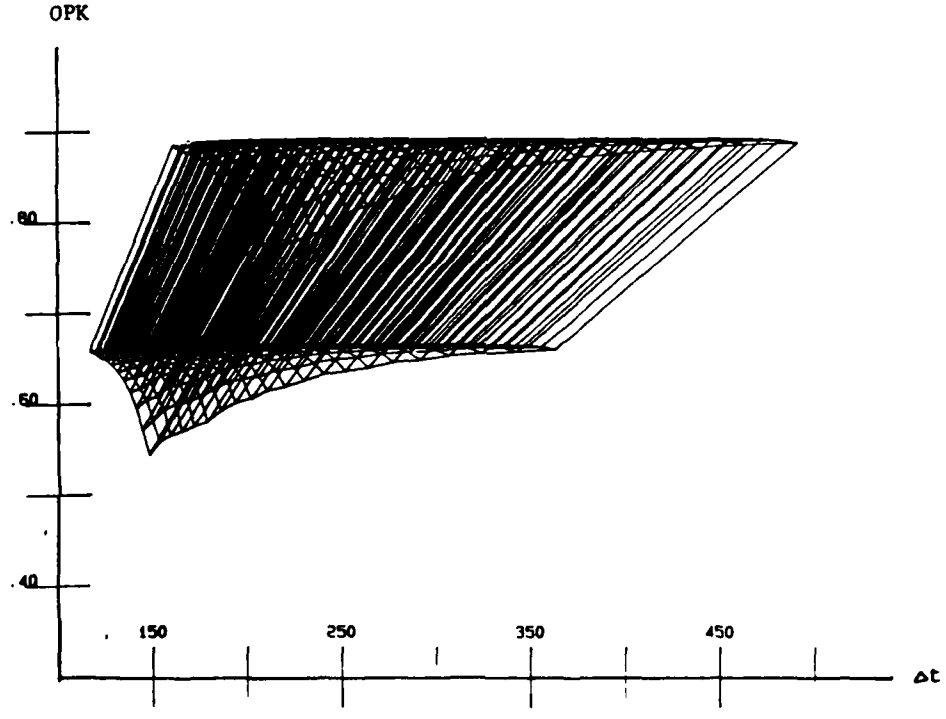


Figure 5.5 System Locus with Double the Data Points

Trunc gives the greatest integer less than or equal to Num, if Num ≥ 0 or the smalllest integer greater than or equal to Num, if Num < 0.

Cothier(1984) generated data from a different system and even if the error involved in the rounding was only slight, it would make a difference. For example, if the number before the round operation in one computer system is 3.5001 and within the round operation 3.5001 + 0.5 = 4.0001 will yield 4 as the rounded number of shots. However, if another computer system has computed the same number as 3.4999 and within the round operation 3.4999+0.5 = 3.9999, then the round operation will yield 3 as the rounded number of shots. In summary, different software and computer systems may treat numbers slightly differently, causing variations in data.

5.3 The One Battery Case

Figures 5.6(a) and 5.6(b) show the system loci for Doctrine 1 (L.S.S.S.) and (L.S.L.S.). Figure 5.6(c) shows the locus of Doctrine 2 in cross-sections which displays the discontinuities in the locus clearly. These loci agree with Cothier's (1984) data. From these loci, Cothier came up with the following results, briefly restated here:

Doctrine 1

• For low target speeds, the window of opportunity is wide and a large number of shots is possible. Moreover, because of the slowness of the target, each SSPK is high. The number of shots is large enough to compensate for the inaccuracy in the observation and the quality of the estimates does not play a fundamental role.

• For high target speeds however, the window of opportunity is narrow, and only a smaller number of shots is possible. The analysis reveals that it is worth spending a longer time to get goood estimates and then fire as may shots as possible on this basis. Since there is only one estimation with no later updating, it should be as accurate as possible. If only few shots are usable, then it is desirable to get better estimates - the overall OPK is higher and costs less because fewer shots were fired.

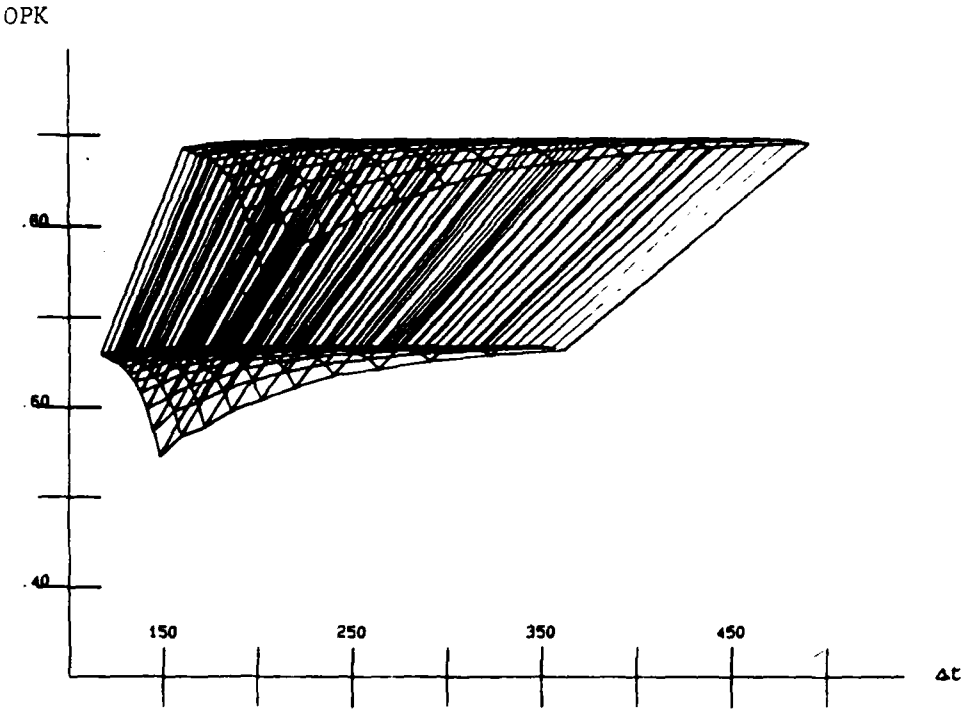


Figure 5.6 (a) Doctrine 1 System Locus

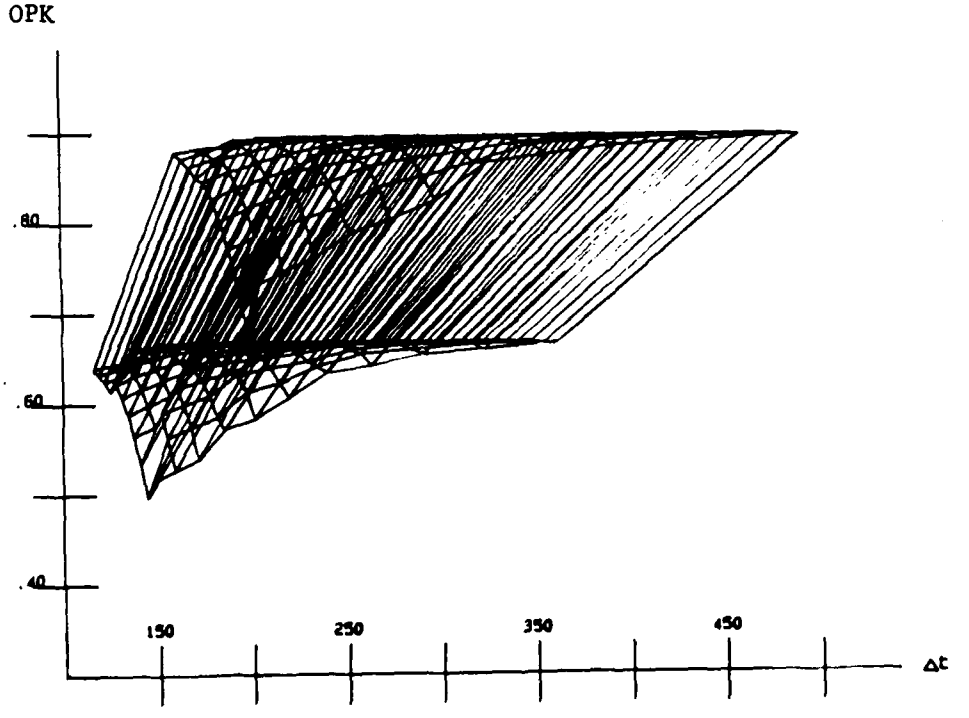


Figure 5.6 (b) Doctrine 2 System Locus

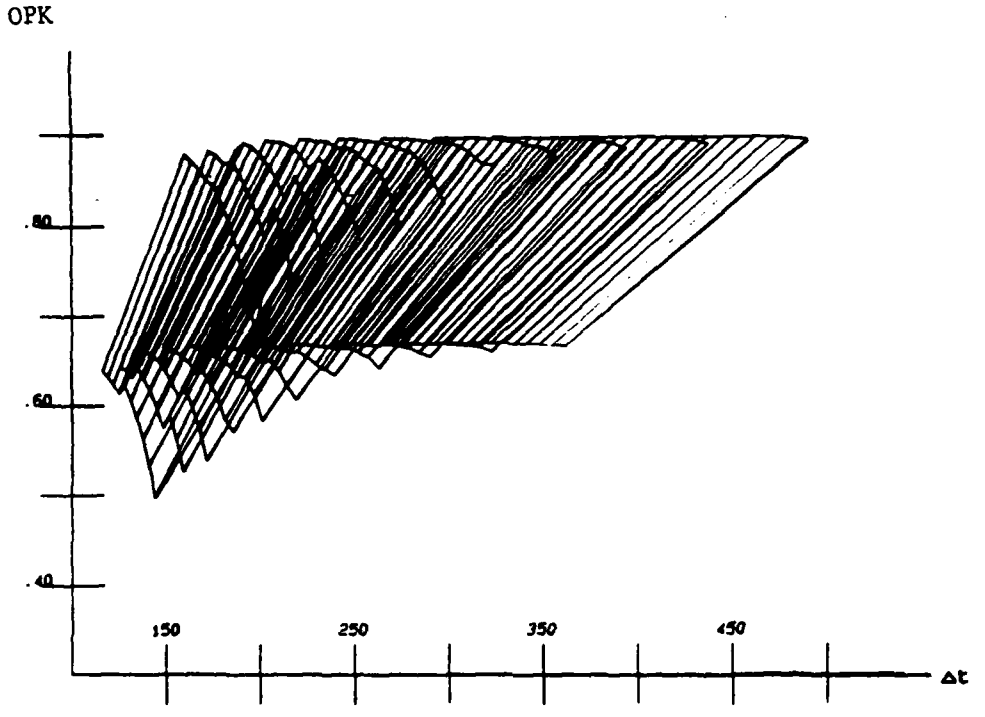


Figure 5.6 (c) Doctrine 2 Systems Locus – Cross Sections

Doctrine 2 (L.S.L.S.)

• A large part of the time available is used for making estimates of the target. As a consequence, the number of possible shots is low.

• For low target speeds, each SSPK is high because of the slow motion of the target, the resulting overall OPK is very high. The quality of the estimates does not play a fundamental role.

• For high target speed, the window of opportunity is narrow, and the number of possible shots is quite small. Each shot is crucial and to forego one of them in order to improve the quality of the estimates and the SSPK has an overall negative effect. Cothier(1984) suggested to reduce the uncertainty up to the point where further reduction reduces the OPKs

Comparison of Doctrines 1 and 2

Figures 5.7 and 5.8 show the different projections of the locus on the attribute planes. In Figure 5.7, the (t**,OPK) plane, the upper limit of $OPK_{max} = 0.8939$ is shown and the the discontinuties of the Doctrine 2 system locus are further emphasized. Figures 5.8(a) and 5.8(b) are indistinguishable, which is logical. The projection is in the (Δt, t**) space, attributes which, together, completely characterize the window of opportunity. The window of opportunity is determined by the geometry of the system which does not change with the doctrine, so the plots should be exactly alike. This projection also shows that t**, the upper limit of the window of opportunity is the same for both doctrines for each value of ω and thus, as Cothier (1984) noted, the t** attribute does not play a critical role in the building of the system loci. Remember that t** is simply the time the target crosses the last point on its path at which the battery can send fire:

$$t^{**} = x(\text{last fire point}) / \omega = K_1 / \omega \qquad (5.4)$$

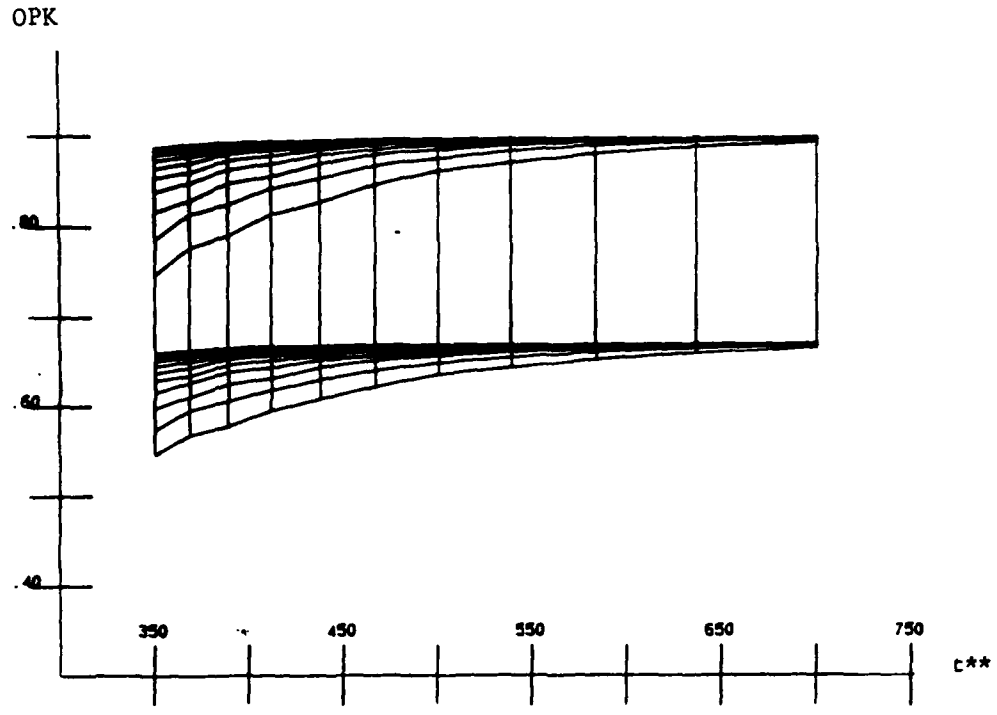


Figure 5.7 (a) Doctrine 1 System Locus - Side View

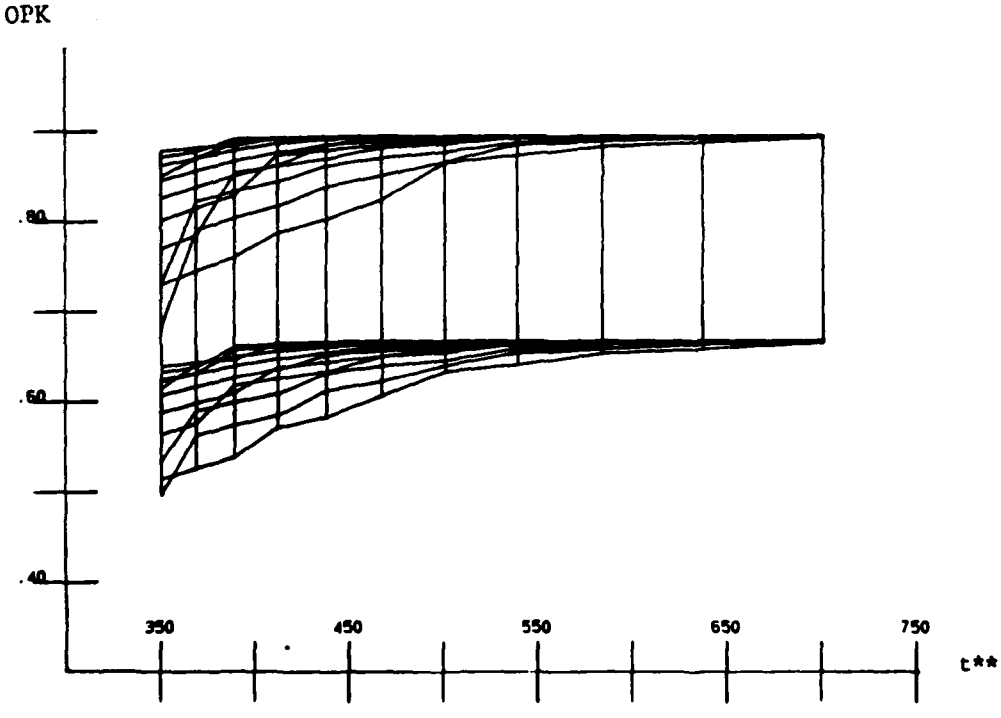


Figure 5.7 (b) Doctrine 2 System Locus - Side View

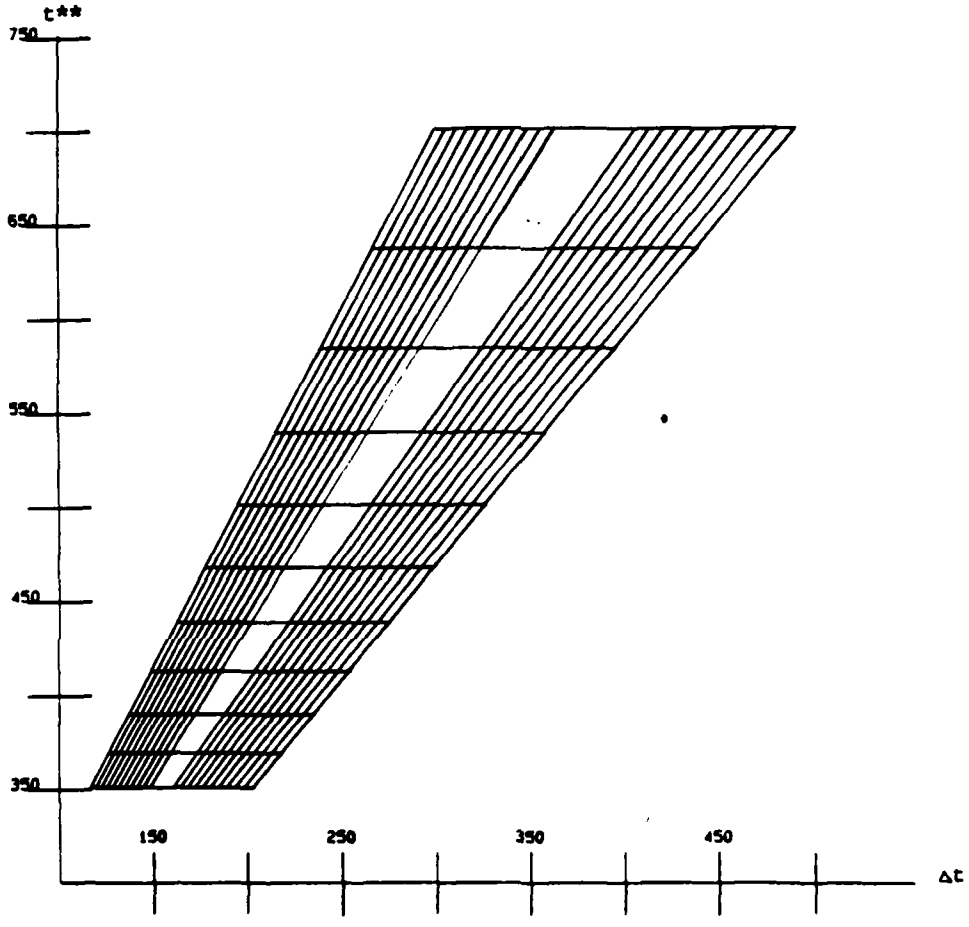


Figure 5.8 (a) Doctrine 1 Systems Locus - Top View

In Figures 5.9(a) and 5.9(b), a projection of the mission locus is shown with the system loci. At low speeds, the intersection of the mission and system loci shows no difference between doctrines. The saturation level is reached because there are many shots in Doctrine 1 and there is a high SSPK in Doctrine 2. But at high speeds, Doctrine 1 is better because Doctrine 2 has discontinuties that result in a lower total volume intersection. Because the target is moving fast, it is better to make one estimate than to waste time making better estimates: the window of opportunity is small. In summary, doctrine 1 works better for high speed targets while doctrine 2 is better for low speed targets since it fires less shots (lower cost) with the same resulting overall OPK as Doctrine 1.

The ratio of the volume of the intersection of the mission and system loci over the volume of the system locus is larger with doctrine 1 than with docrtine 2. The discontinuties result in a lower intersection volume for doctrine 2:

$$E_1 = E(\text{Doctrine1 (1battery)}) \approx 0.55$$

$$E_2 = E(\text{Doctrine 2 (1 battery)}) \approx 0.50$$

As a final demonstration of the graphics program, Figures 5.10(a) and 5.10(b) show the orthographic projection of the system loci rotated in three dimensions. The loci have been rotated 25° around the x axis, 25° around the y axis, and 25° around the z axis. All rotations are computed under the right-hand rule with the center of rotation at the center point of the locus. The center point is computed by finding the maximum and minimum point in each dimension and computing the middle value between the difference of the two. The following formula is used:

Center point (x,y,z)=

$$(x_{min}+[x_{max}-x_{min}]/2, y_{min}+[y_{max}-y_{min}]/2, z_{min}+[z_{max}-z_{min}]/2)$$

A unit vector showing the rotation of the axes is shown in the lower left hand corner of the figures. Each locus is twisted in a way the previous 2-D projections did not suggest. The similarities of the loci under each doctrine are apparent at low speeds, while the

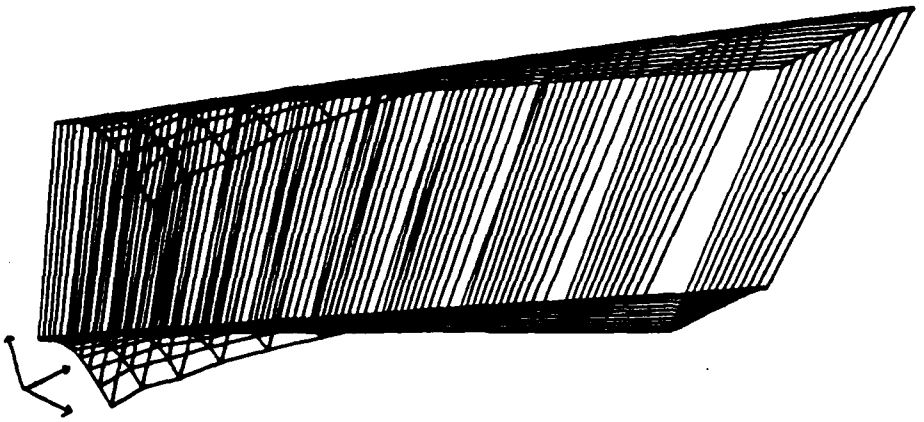


Figure 5.10 (a) Dotrine 1 System Locus Rotated 25° around x, y, and z axes

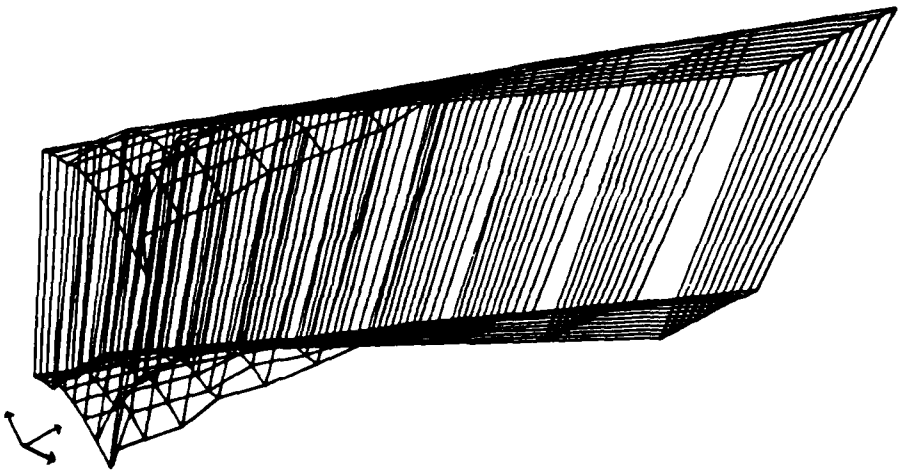


Figure 5.10 (b) Dotrine 2 System Locus Rotated 25° around x, y, and z axes

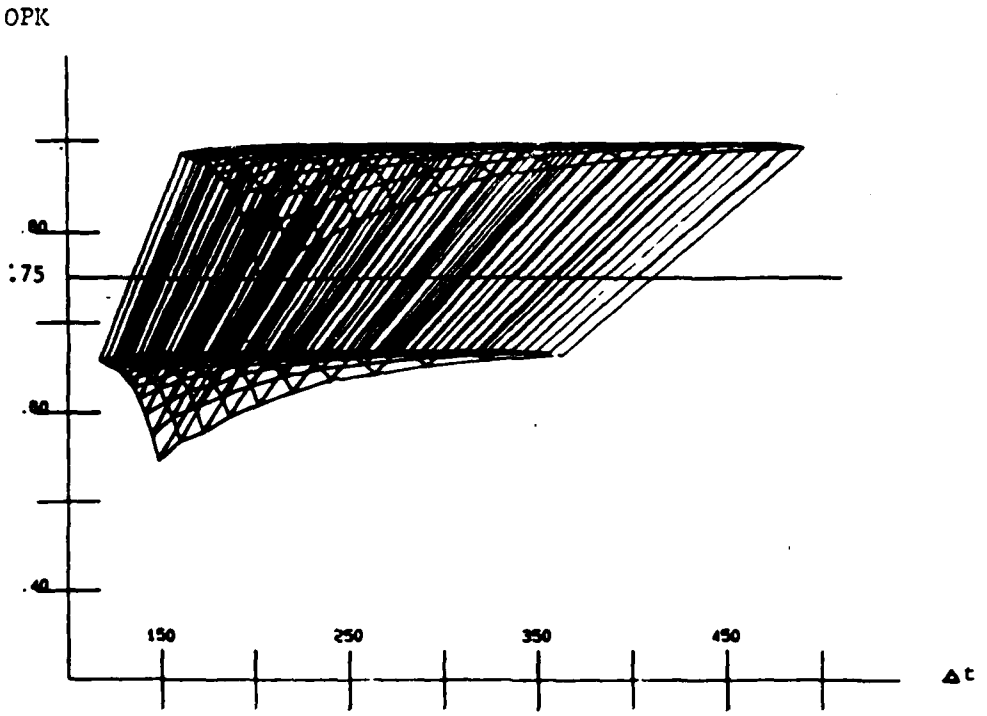


Figure 5.9 (a) Doctrine 1 System and Mission Loci

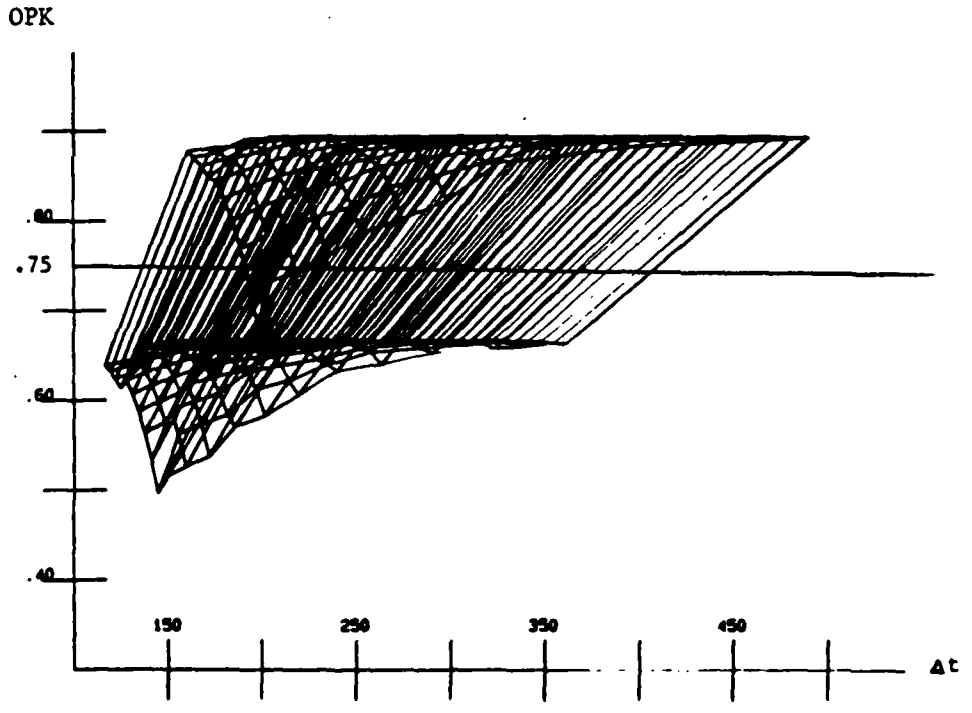


Figure 5.9 (b) Doctrine 2 System and Mission Loci

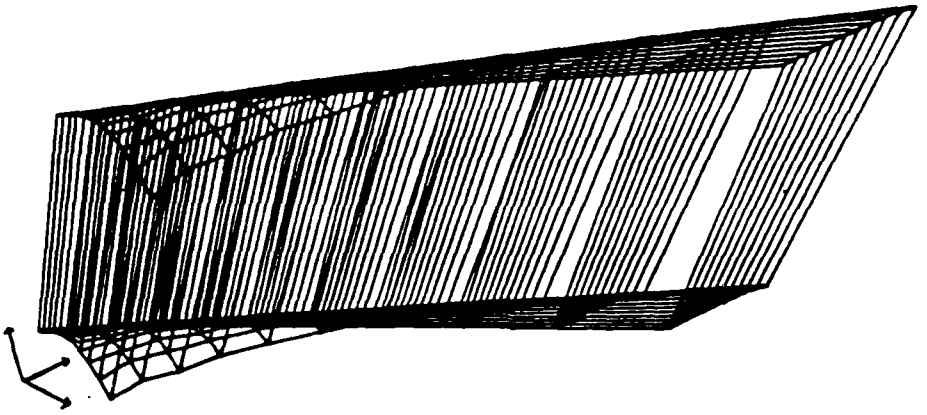


Figure 5.10 (a) Dotrine 1 System Locus Rotated 25° around x, y, and z axes

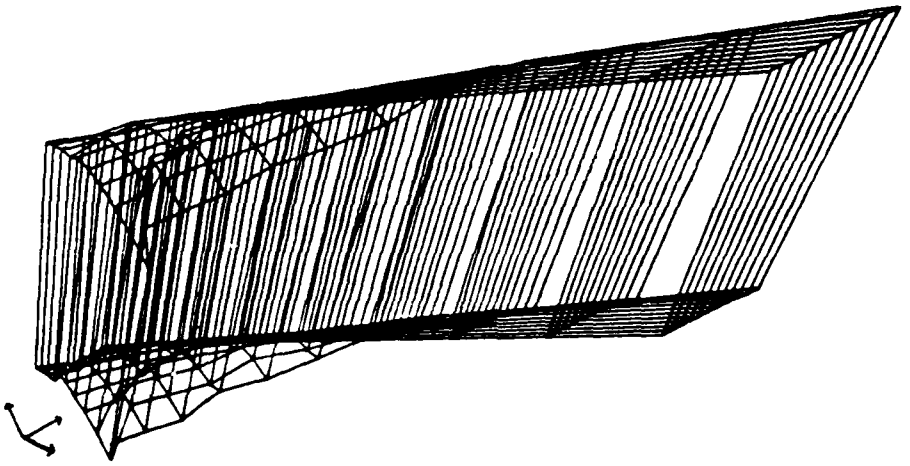


Figure 5.10 (b) Dotrine 2 System Locus Rotated 25° around x, y, and z axes

discontinuities of the doctrine 2 system locus can be seen clearly. Rotating the loci in three dimensions will enable one to observe properties that can not be seen in two dimensional viewing.

5.4 The Two Battery Case

When a second battery is added to the system configuration, the window of opportunity is changes depending on the option. Cothier (1984) considered the two-battery case under doctrine 1. This case will be repeated here along with the two-battery case under doctrine 2.

Doctrine 1 (L.S.S.S.), Option 1 (no coordination)

Figure 5.11(b) shows the system locus under doctrine 1 and option 1. The longer length along the Δt axis indicates the increase in the width of the window of opportunity. Compare this the the system locus under doctrine 1, Figure 5.9(a). At low speeds, the OPKs for both systems are the same which is at the saturation level, but at high speeds, the overall OPKs are higher in the two-battery case. Because the addition of a second battery widens the window of opportunity and allows for more shots, the overall OPKs increase.

Doctrine 1, Option 2 (coordination)

Figure 5.11(b) shows the system locus under doctrine 1 and option 2. The width of the window of opportunity decreases because it is now that of battery 2 in order to have coordinated fire from both batteries at the same time. Note also that the shape of the cross sections has changed. As the angle β varies, there is no change in the window of opportunity. This indicates that for low speeds, the OPK only depends on the probability p. A closer look shows that the OPK is at its saturation level, and thus any improvement by using a larger β in the estimation has no effect. At high speeds, however, the OPK does increase with β. The forward observer uses the time before the target enters the window of opportunity to make an accurate observation.

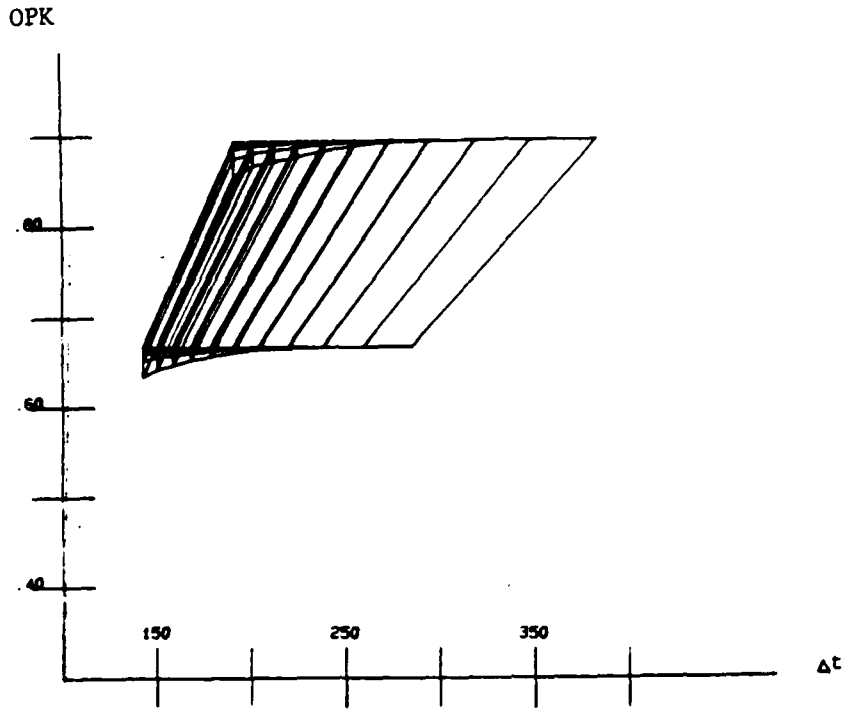


Figure 5.11 (a) Doctrine 1, Option 2 (coordinated batteries) System Locus

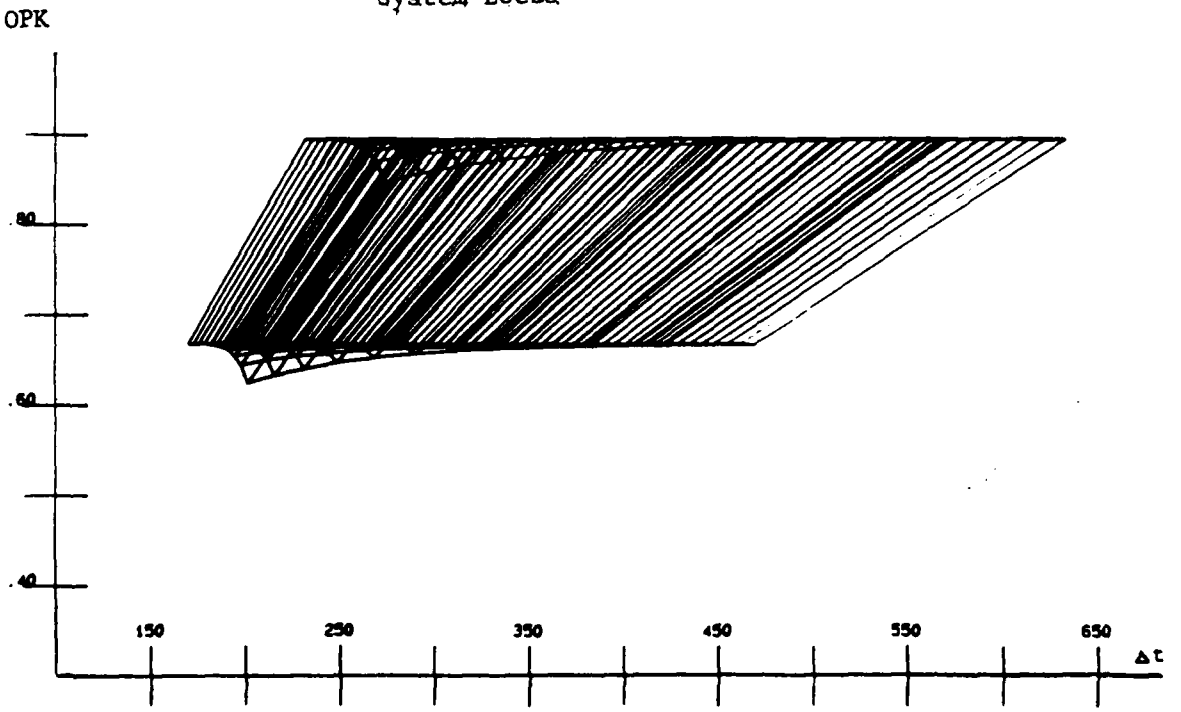


Figure 5.11 (b) Doctrine 1, Option 1 (uncoordinated batteries) System Locus

Comparision of Option 1 and Option 2

Although the length of the option 2 system locus is smaller than that of option 1 (Δt has been reduced), the OPKs are almost the same. Therefore, even with a small window of opportunity, the same OPKs can be achieved. Moreover, the ratio of the volume of the intersection of the mission locus and the system locus over the volume of the system locus is the same value for both loci:

$$E_3 = E(\text{Doctrine 1, Option 1})$$

$$E_4 = E(\text{Doctrine 1, Option 2})$$

$$E_3 \approx E_4 \approx 0.6$$

First notice that the effectiveness value is 0.05 higher than the effectiveness of the doctrine 1 system locus. Because in the two battery case the effectiveness is the same, option 2, coordination with two batteries under doctrine 1, has a higher quality. Less shots are fired under option 2 and thus there is a lower cost for the same results. Moreover, the survivability of the system is increased because the target will not start countermeasurs during the time battery B_1 is waiting, while in option 1, the countermeasures may begin as soon as battery B_1 starts firing. In summary, the time of the window of opportunity is better managed in option 2; a larger window of opportunity does not imply that the option is of higher quality.

Doctrine 2, Option1 (uncoordinated)

Figure 5.12(b) shows the system locus for doctrine 2, option 1. The width of the window of opportunity is the same as in doctrine 1, option 1, but in this case, there are discontinuities in the locus. Compared to doctrine 2 in the one battery case (the locus with the discontinuities), there is a clear improvement. At low speeds, the cross-sections are the same. At high speeds however, with small angles of observation β, doctrine 2 , option 1

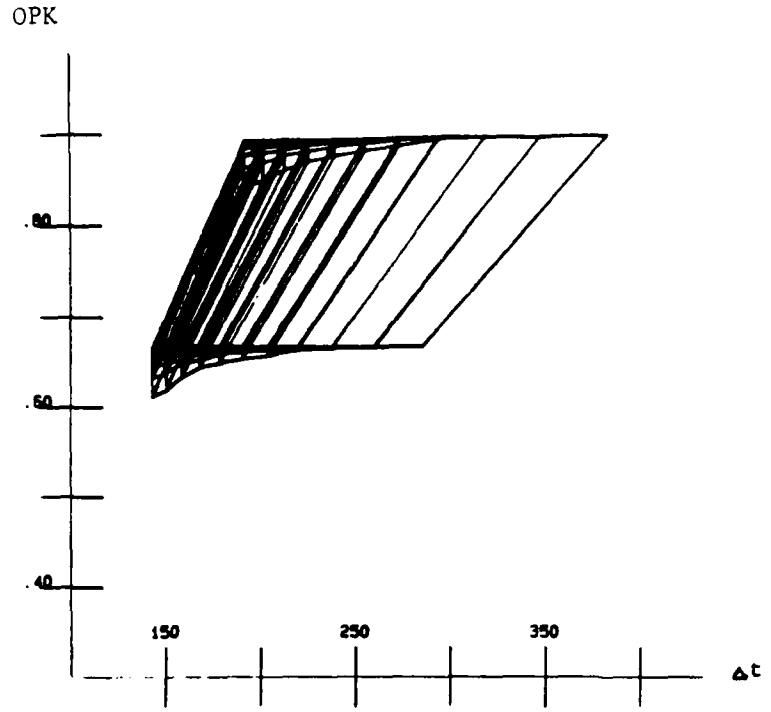


Figure 5.12 (a) Doctrine 2, Option 2 (coordinated batteries) System Locus

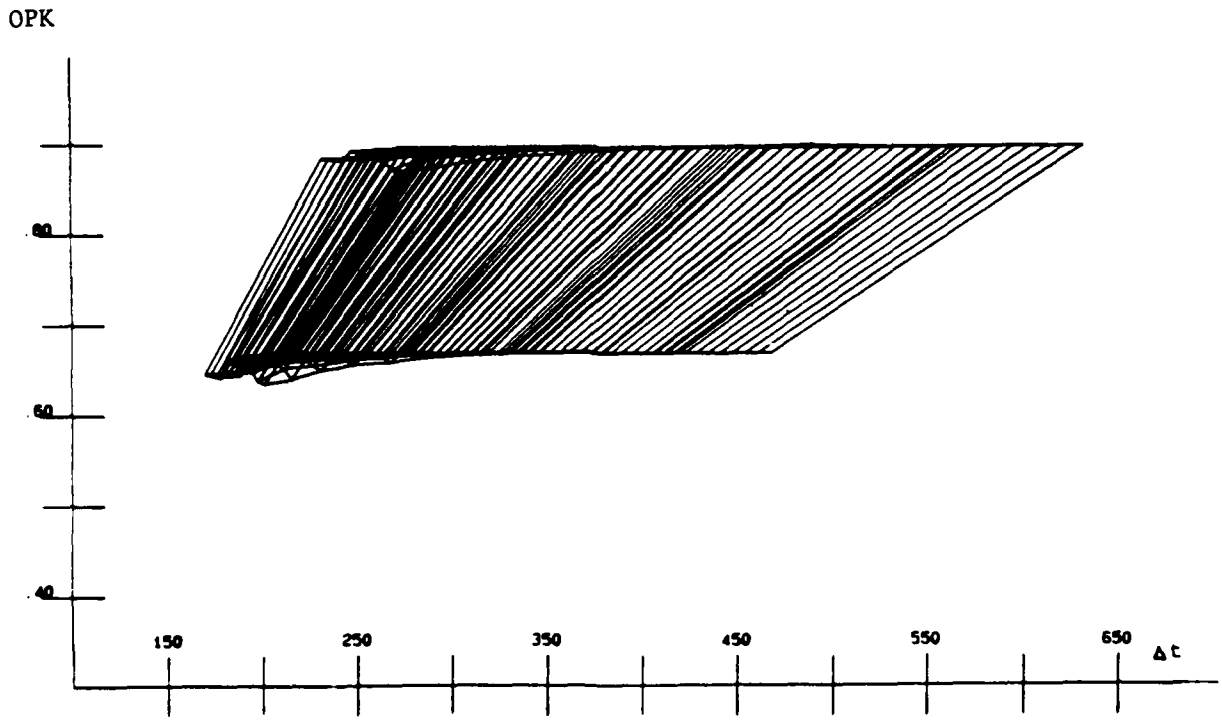


Figure 5.12 (b) Doctrine 2, Option 1 (uncoordinated batteries) System Locus

achieves higher OPKs while at large angles of observation β, doctrine 1, option 1 achieves higher OPKs. The quality of doctrine 2, option 1 may be higher, if the observations angles are small. First, doctrine 2 requires less shots and thus costs less. Second, using a small angle of observation means that the forward observer will be making the observation in less time and thus is less vulnerable to be attacked. The discontinuities in the locus occur at large angles of observation because the window of opportunity is small at high speeds and if a long observation is done, very few shots can be fired. This causes a drastic decrease in the overall OPKs. Note that in the case of doctrine 2, 1 battery, the discontinuities occured at smaller βs. The effect of a second battery smooths these out by firing a shot when the other battery has to wait too long for an estimate from the forward observer.

Doctrine 2, Option 2 (coordinated)

Figure 5.12(a) shows the system locus for doctrine 2, option 2 which is similar to the doctrine 1, option 2 system locus in Figure 5.11(a). The width of the window of opportunity is the same which is logical because the geometry does not change under docrtines. The discontinuties from the 1 battery doctrine 2 case have been smoothed out. The two coordinated shots fired with less uncertainy increases the OPKs even at small angles of observation. At low speeds, there is no difference between the two doctrines under option 2. At high speeds, however, the angle of observation, β, has more of an effect on the OPK in doctrine 2. At small angles of observation with the target moving quickly, the OPK is lower than that of doctrine 1.

Comparasons of Options 1 and 2 under Doctrines 1 and 2

Most of the same comparision can be drawn between doctrine 2, option 1 and 2, that were drawn between docrtine 1, option 1 and 2. The interesting comparision is between doctrines in the two battery case. As stated above, doctrine 2 exemplifies a better management of time and the effectiveness value under both doctrines is the same. Doctrine 1, option 2 may be of higher quality than that of Doctrine 2, option 2 for the following reason. Although more shots are fired under doctrine 1, there is only one observation made.

In summary, doctrine 1, option 2, may be best set of operating rules because

- the quality of the option is high since there is a better management of time.
- less shots are fired than the case of uncoordinated fire.
- the effectiveness is high (the shape of the locus is almost the same).

5.5 Global Measure of Effectiveness

The final step of system effectiveness methodology is to compute a gloabal measure of effectiveness for the TACFIRE system evaluated. As an example, consider the system to consist of case one to four. The gobal measure is computed with the partial measures used as arguments of a utility function. The utility function has the form:

$$u(E_1, E_2, E_3, E_4) = E_1^{\gamma 1} \cdot E_2^{\gamma 2} \cdot E_3^{\gamma 3} \cdot E_4^{\gamma 4}$$

with

$$\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1$$

From Cothier(1984), the following values were chosen:

$$\gamma_1 = \gamma_2 = 0.3$$

$$\gamma_3 = 0.15$$

$$\gamma_4 = 0.25$$

The coeffecients γ_1 and γ_2 are set equal , if one does not judge, a priori, the adequacy

of the two different doctrines with respect to one another. However, they are larger than γ_3 or γ_4 because these coefficients refer to the two-battery system which is more expensive. And because the quality of option 2, doctrine 1, is better than the other three, γ_4 is higher than γ_3. Finally, the gobal measure of effectiveness is:

$$E = (0.55)^{0.3} \cdot (0.50)^{0.3} \cdot (0.60)^{0.15} \cdot (0.60)^{0.25}$$

$$E = 0.553$$

Note that this is an example of just one measure of effectiveness; others exist but have not been explored in this thesis.

Then next chapter will introduce the system with a second forward observer, thus making the system model more realistic. Results will be compared to the analysis in this chapter.

CHAPTER VI

ANALYSIS OF THE DOCTRINES AND OPTIONS USING THE GRAPHICS SYSTEM: TWO OBSERVERS CASE

6.1 Introduction

To make the model of the TACFIRE system more realistic, a second forward observer has been introduced. For simplicity, the second FO will have the same y coordincate as the first FO. Only the distance between them along the x axis will change. Loci have been constructed using systems effectiveness analysis and the graphics system. These loci will be analyzed to asses the two FO system loci separated at various distances. Note that only the system under Doctrine 2 (L.S.S.S.) is being considered. Doctrine 1 (L.S.L.S.) assumes that there is one observation and all the shots thereafter are fired from information based on this observation. Thus, a second FO would be ignored by the rest of the system and the system locus would not change.

First, this chapter will present the two FO system with a study of the effect of the distance between the two FOs on the system effectiveness. Second, the case of the second FO at the same coordinates as the first FO (d=0, $x_{FO1} = x_{FO2}$, $y_{FO1}=y_{FO2}$) is analyzed. Third, the effect of a 'blind spot' - a length along the threat trajectory that cannot be observed by either FO_1 or FO_2 - is presented. Finally, the two battery case with two FOs separated by distance d=1.0 miles is analyzed under option 1(uncoordinated) and option 2 (coordinated).

6.2 Two Forward Observers Separated by Distance d: The One Battery Case under Doctrine 2

Figures 6.1(a) - 6.1(d) show the changing geometry as the distance between the two FOs goes from 2.0 miles down to 0.05 miles. The window of opportunity is determined by the constraints of battery B_1. Figures 6.3(a) - 6.3(i) show the system loci for the system whose changing geometry is specified in Figures 6.1(a) - 6.1(d).

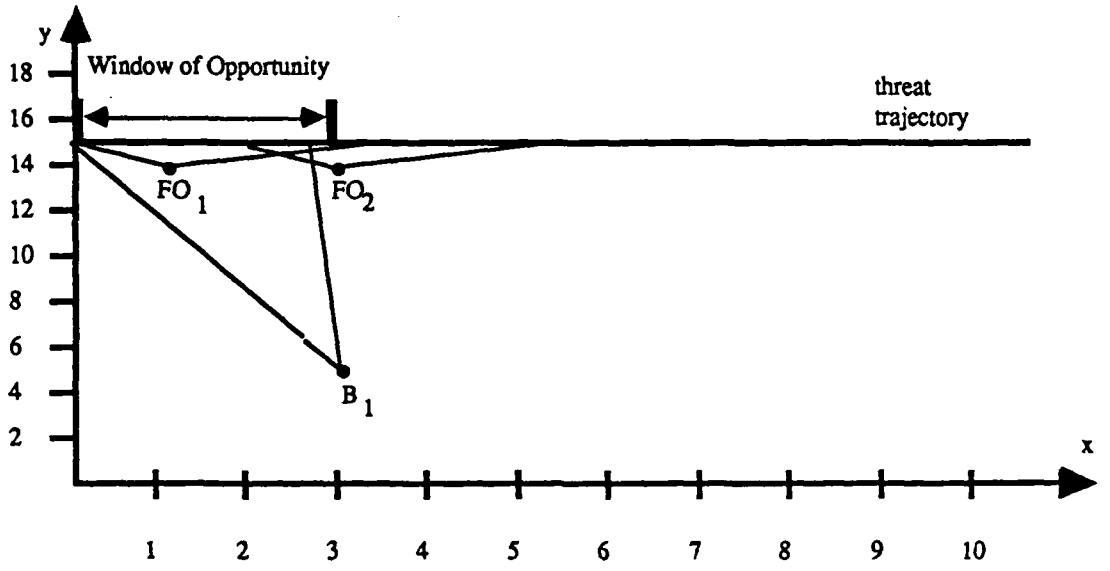


Figure 6.1(a) Two FOs separated by distance d=2.0 miles

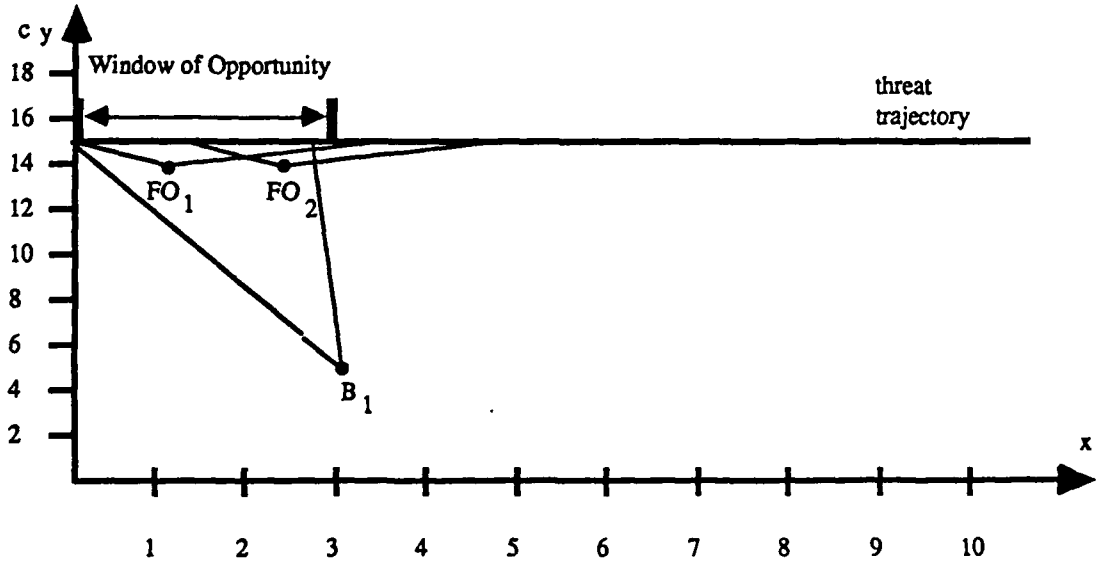


Figure 6.1 (b) Two FOs separated by distance d=1.5 miles

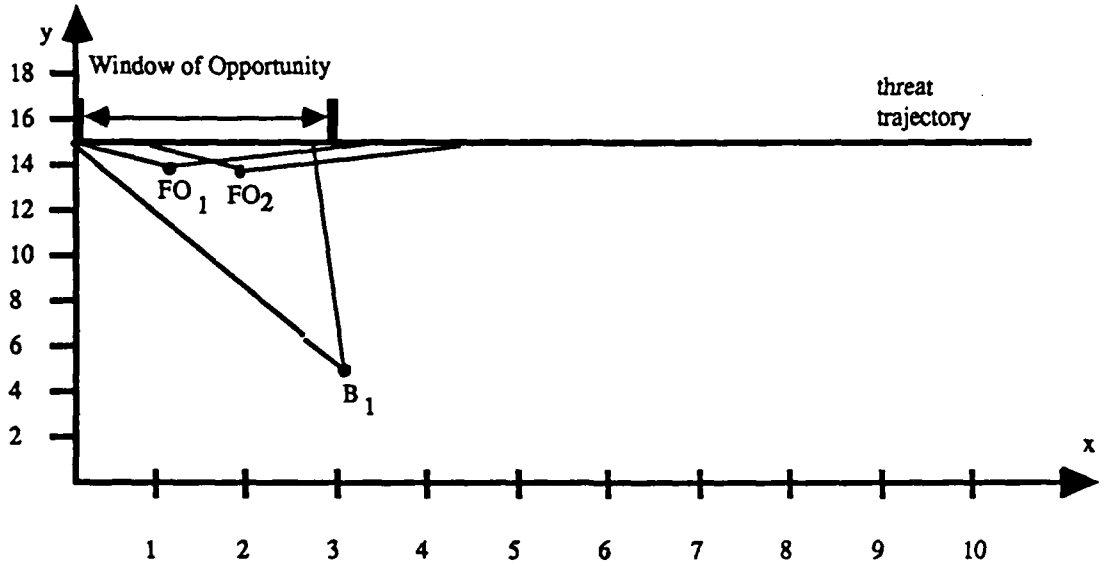


Figure 6.1(c) Two FOs separated by distance d=1.0 miles

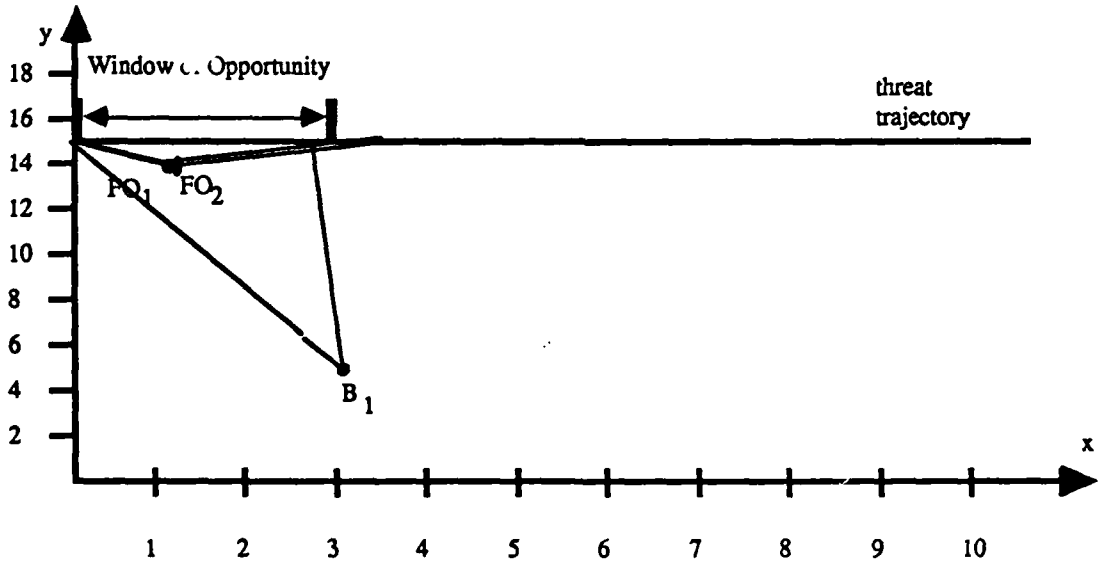


Figure 6.1(d) Two FOs separated by distance d=0.05 miles

Consider the first observer fixed and the second observer moving a distance d along the x axis from x_{FO1} so that $x_{FO2} = x_{FO1} + d$. As this distance changes, the number of shots the system fires also changes. The number of shots resulting from observations made by FO_1 does not change, but the number of shots resulting from FO_2 does, as the second forward observer is able to make more observations. Figure 6.2 shows a plot of the number of shots due to the second FO versus the distance between the forward observers. A sample was taken at ω=17 mph, $\beta = 5°$, and p = 0.95. The number of shots plotted is the sum of the shots from FO_2 along path #1 to #4:

$$\text{Number of shots plotted} = \sum_{i=1}^{4} \text{number of shots (i)}$$

The number of shots is directly related to the OPKs; the more shots there are on a given path i, the higher the OPK becomes for that path and the higher the overall OPK becomes. Figure 6.2 predicts where there will be changes in the system locus as the distance changes. Every time there is a change in the number of shots, there should be a change in the system locus.

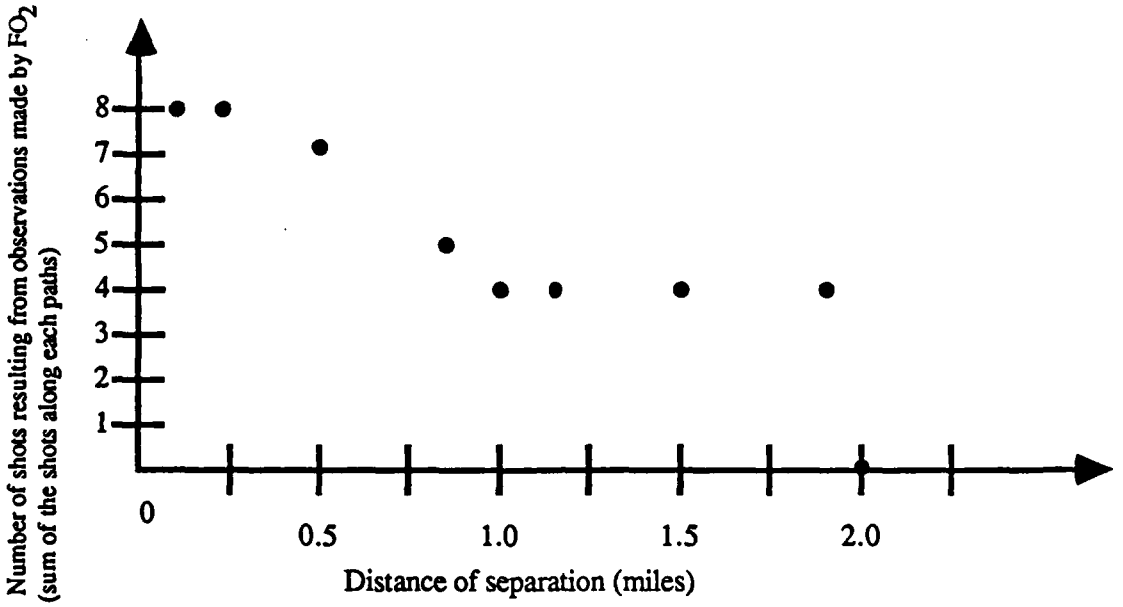


Figure 6.2 Number of shots vs. the separation distance d

First, study Figures 6.1(a) and 6.3(a). The system locus is exactly the same as the system locus for the one FO system, the case of one battery under Doctrine 2. Although the second FO makes an observation within the window of opportunity, the battery does not have enough time to make a shot based on this estimate. The impact time would fall outside the window of opportunity. Therefore, the system locus does not change because the number of shots are the same as in the one FO case. The second FO is too far out to have an effect on the system.

Figure 6.3(b) shows the sytem locus with the separation of the two FOs at distance d=1.8 miles. With the two FOs just 0.2 miles closer, the system locus changes. The values of the overall OPKs become higher for almost all values of β and the discontinuities are substantially less sharp. At this distance, an extra shot occurs at all speeds of the threat. As before, if the observation takes too long and an extra shot is missed, the effect will be negative on the overall OPK which results in the discontinuities seen. Figures 6.3(c), 6.3(d), and 6.3(e) are virtually the same as Figure 6.3(b). The number of shots for each cross section is the same. As the distance between the FOs is decreased, there is not enough time to allow for another shot and thus, the overall OPKs do not change.

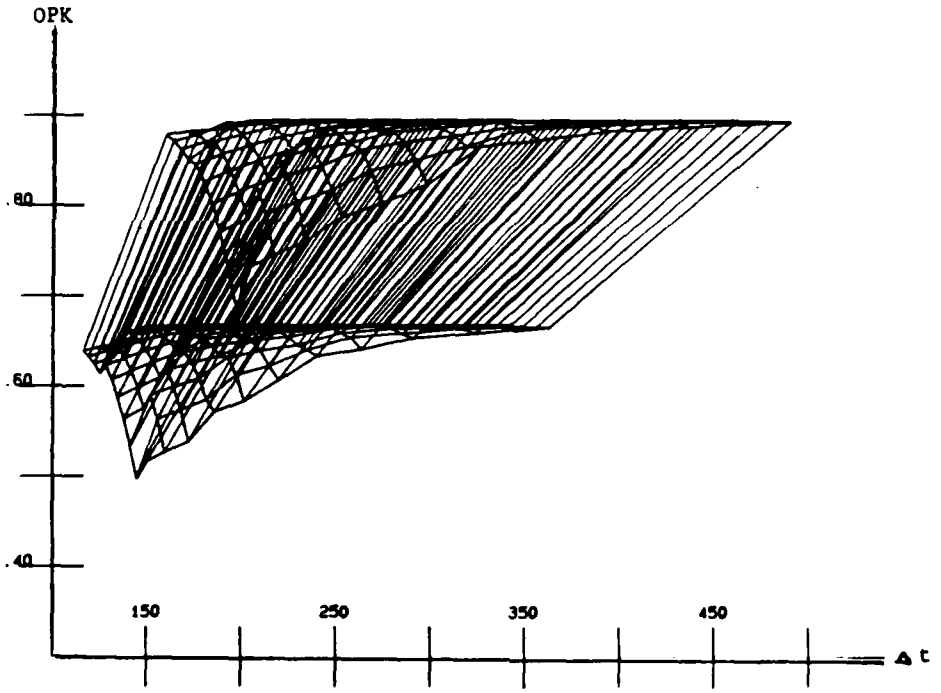


Figure 6.3 (a) Two FOs Separated by Distance 2.0 miles

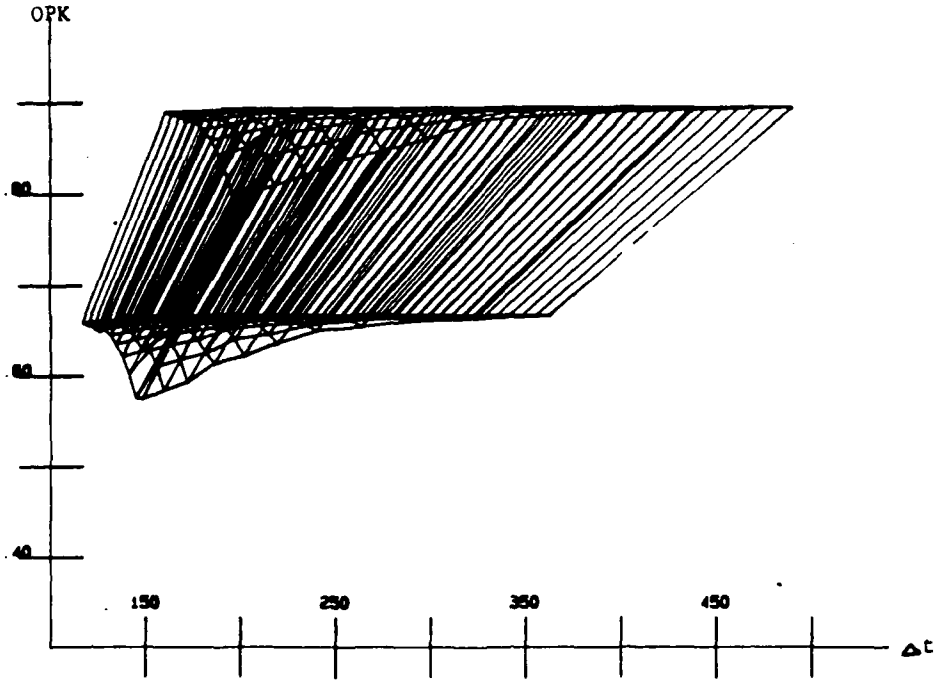


Figure 6.3 (b) Two FOs Separated by Distance d=1.8 miles

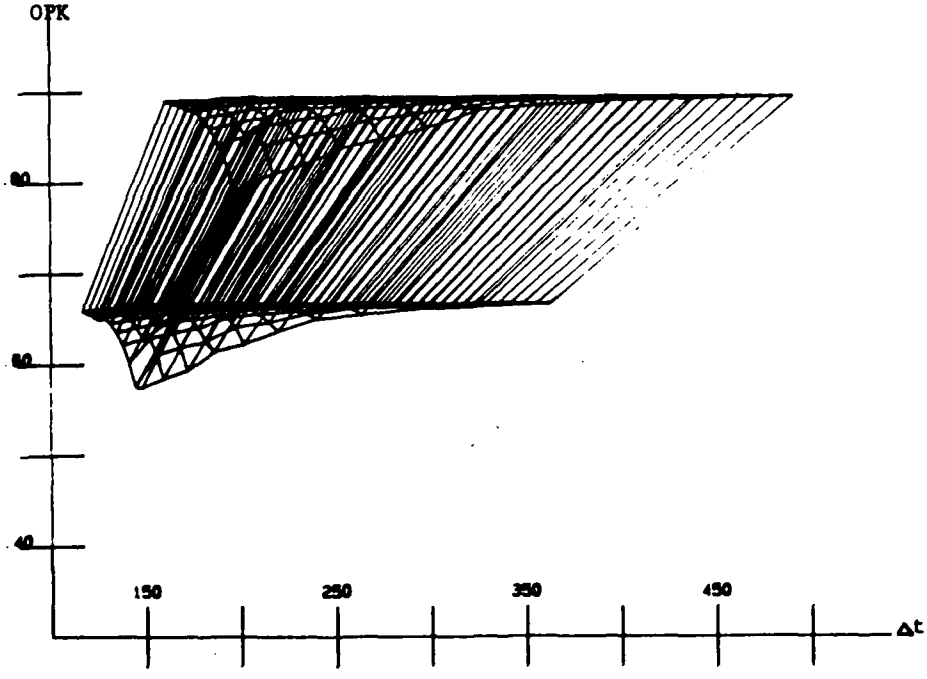


Figure 6.3 (c) Two FOs Separated by Distance d=1.5 miles

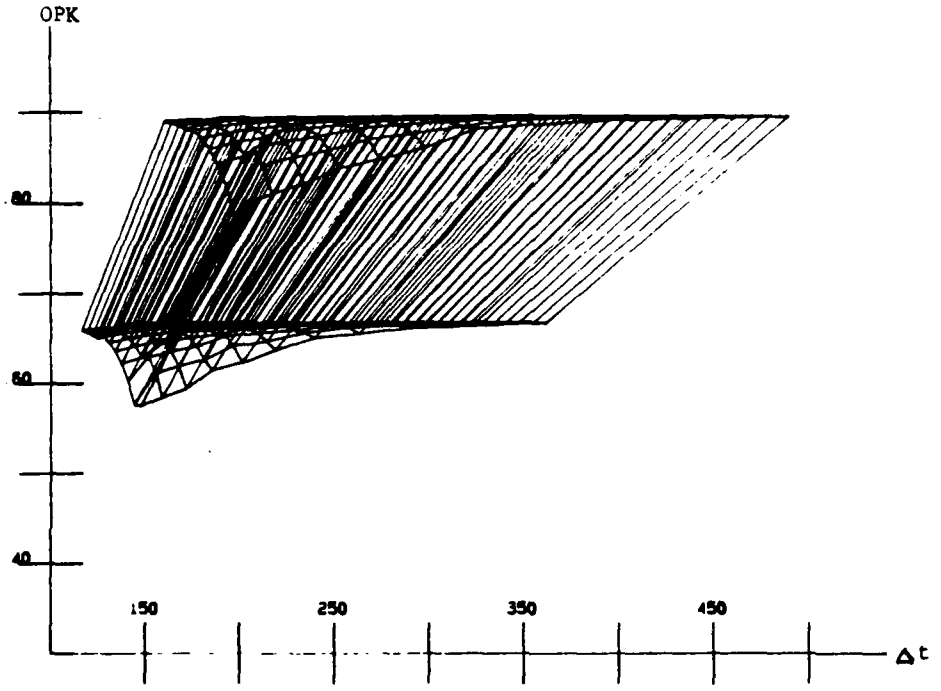


Figure 6.3 (d) Two FOs Separated by Distance d = 1.2 miles

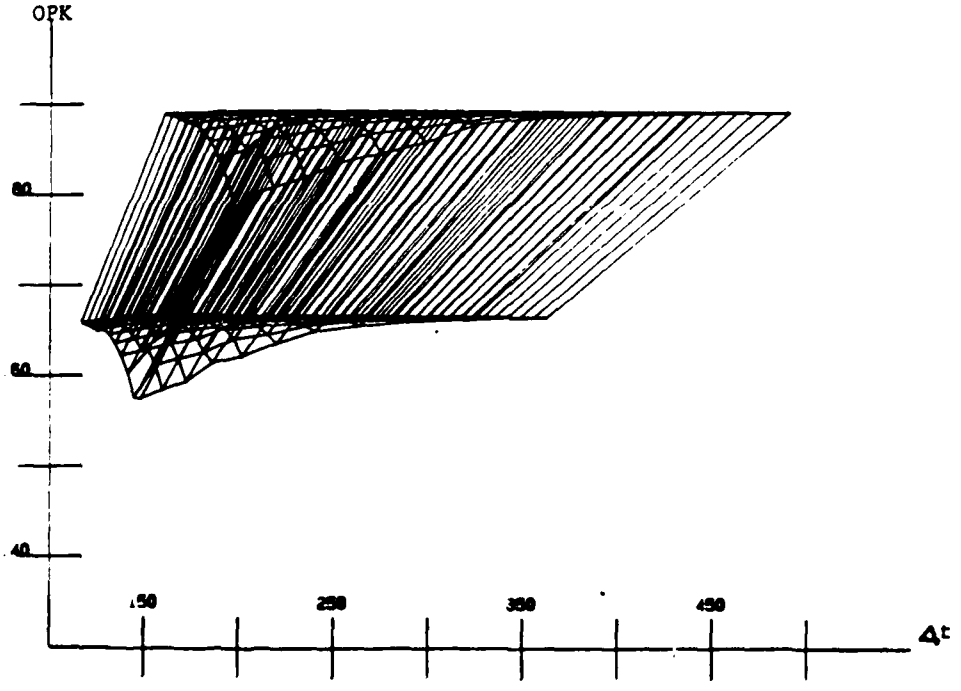


Figure 6.3 (e) Two FOs Separated by Distance d = 1.0 miles

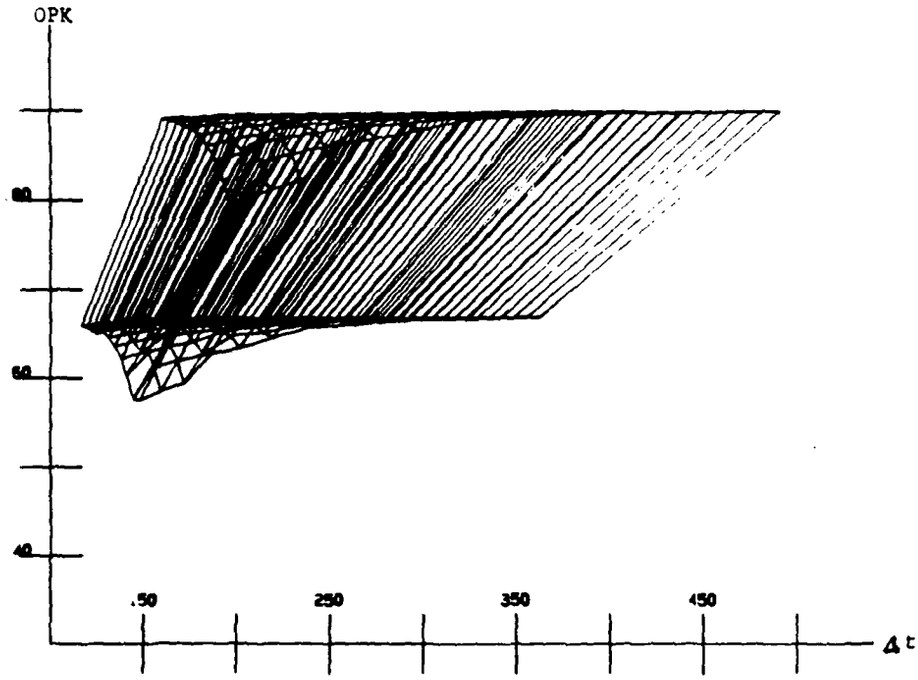


Figure 6.3 (f) Two FOs Separated by Distance d = 0.8 miles

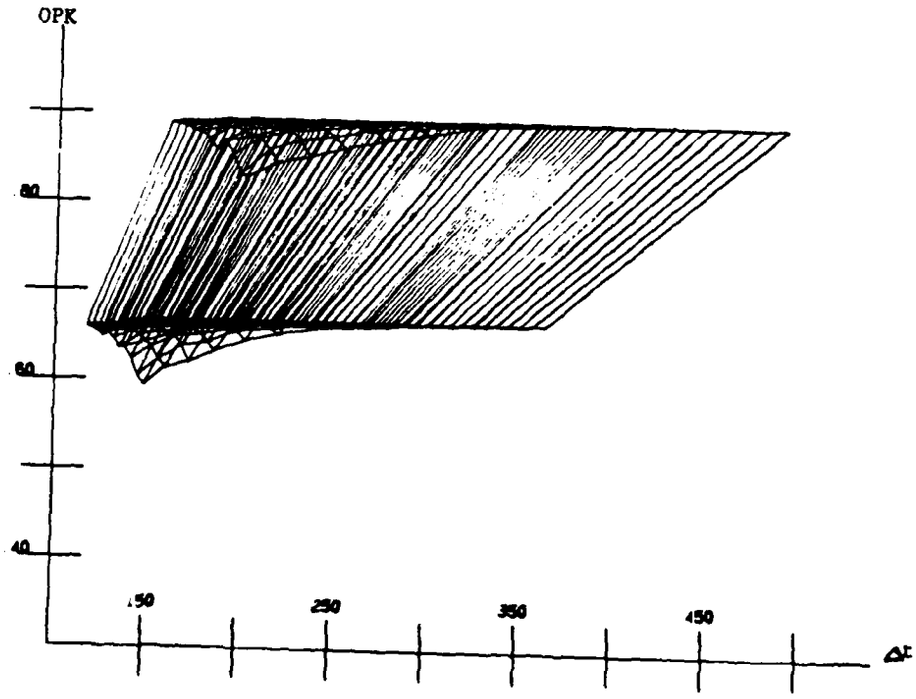


Figure 6.3 (g) Two FOs Separated by Distance d = 0.5 miles

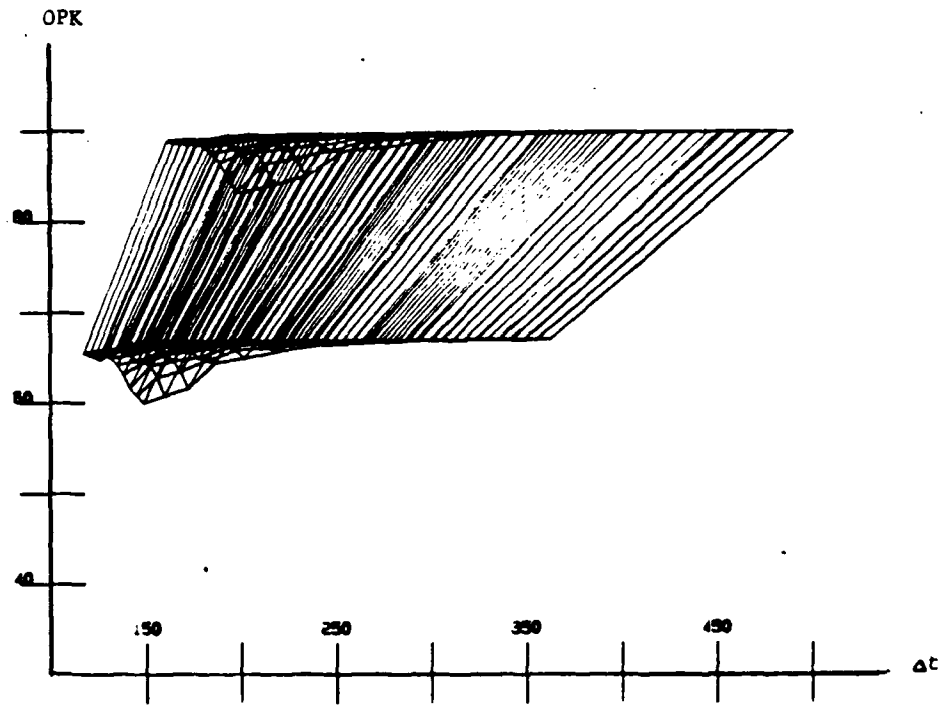


Figure 6.3 (h) Two FOs Separated by Distance d = 0.2 miles

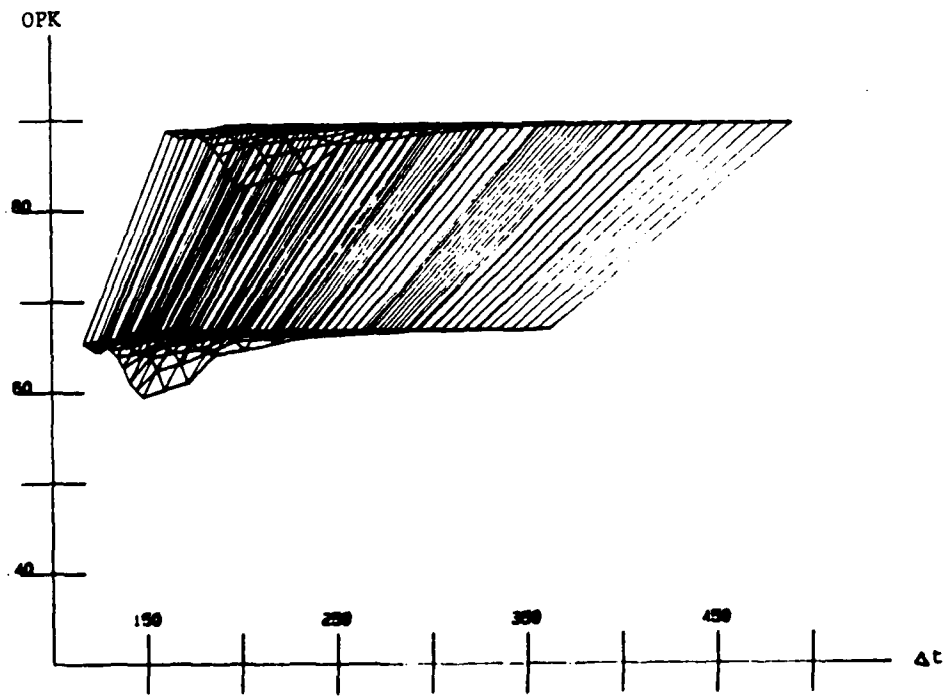


Figure 6.3 (i) Two FOs Separated by Distance d = 0.05 miles

In Figure 6.3(f), the two FOs are separated by 0.8 miles. At this distance, there is a change in the system locus. Notice, however, that at high speeds, the system locus is the same as in Figures 6.3(b)-6.3(e). The locus at ω = 17 mph to ω = 10 mph changes - there are higher overal OPKs. This is a result of an extra shot being fired at these lower speeds, but at high speeds, the threat moves too fast for the system to get an extra shot. In Figure 6.3(g) at distance d=0.5 miles, an extra shot can be fired at high speeds, and the rest of the system locus changes.

Note that for small distances of separation, there must be a wait period for the battery to reload. This is the result of two estimates being made very close in time and commanding the battery to shoot at nearly the same time. For example, the battery has received an estimate from FO_1 and has just fired when an estimate from FO_2 is received. The battery must reload before it can act on this estimate. Thus, two shots are fired consecutively, each based on a different estimate. At any distance lower than 0.2 (which is determined by the equation **reload time * ω_{max} = 0.2 miles**), the system acts in this manner. Figures 6.3(h) and 6.3(i) are exactly alike because the number and time of the shots is exactly the same. However, as the two FOs become closer, the time to wait for the battery to act on the second observation becomes longer.

6.3 Special Cases

A number of special cases exist in the two observer situation. This includes the case of two FOs at the same coordinates in which the information must be fused and the case of a 'blind spot' between the first and second observer where Doctrin 1 (L.S.L.S.) must be applied temporarily.

6.3.1 Two FOs Co-located

Figures 6.4 (a) - 6.4(i) show system loci with the two FOs at the same coordinates ($x_{FO1} = x_{FO2}$, $y_{FO1} = y_{FO2}$). For this case, all the constraints and uncertainties of the second forward observer are the same as that of the first observer. Here, everything is the same except the uncertainties in the second forward observer. Thus, the only thing different

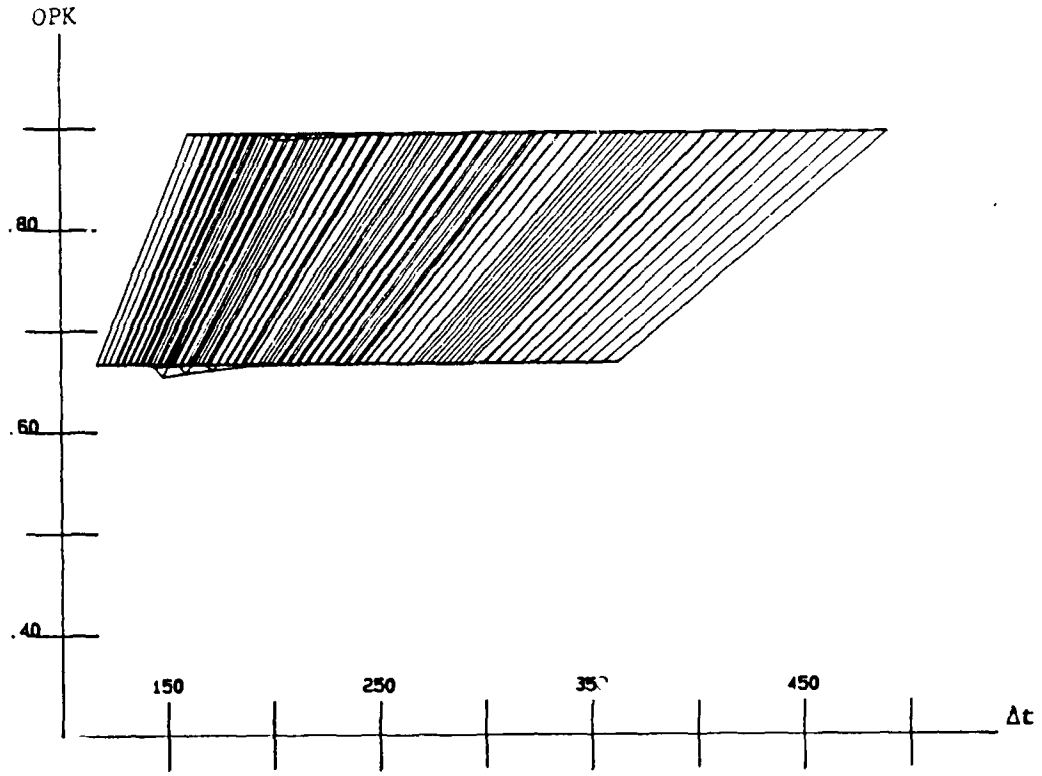


Figure 6.4 (a) Distance = 0 with half of both uncertainties

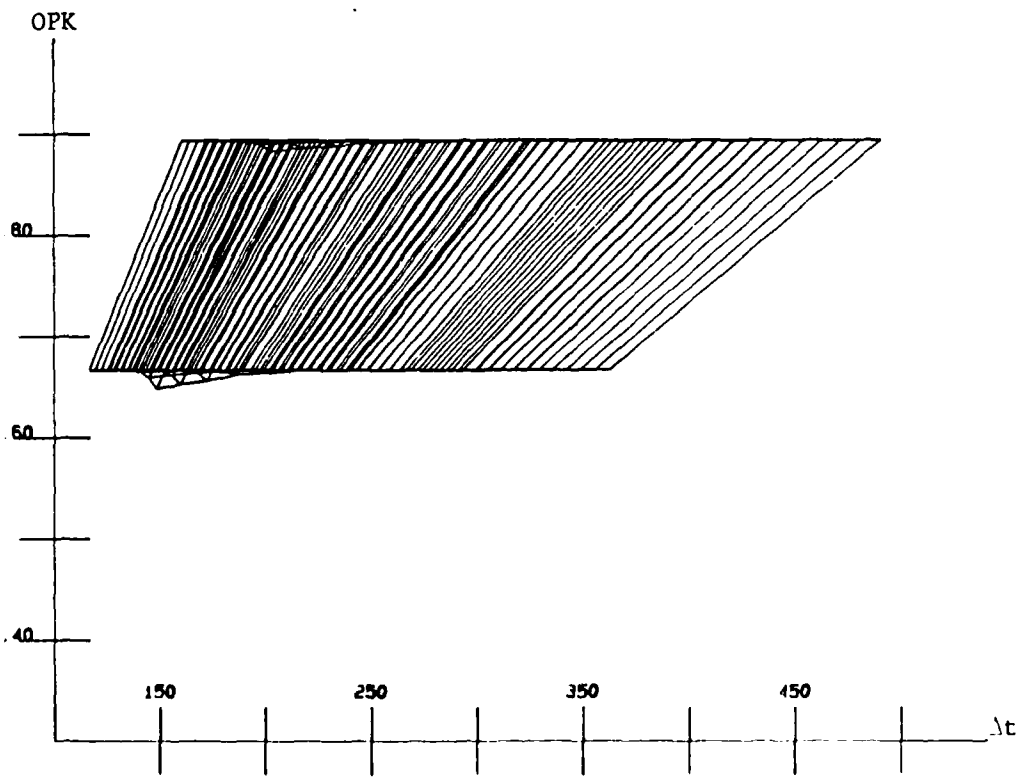
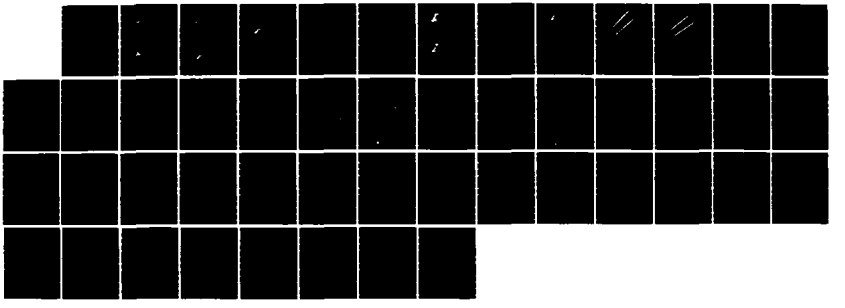
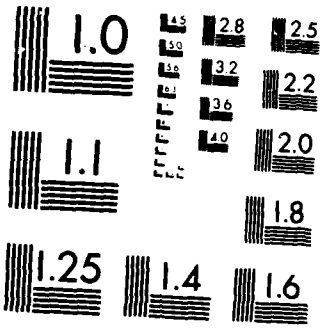


Figure 6.4 (b) Distance = 0 with half the uncertainty in δβ

AD-A173 546 COMPUTER GRAPHICS FOR SYSTEM EFFECTIVENESS ANALYSIS(U) 2/2
MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR
INFORMATION AND DECISION SYSTEMS C M BOHNER MAY 86
UNCLASSIFIED LIDS-TH-1573 N00014-84-K-0519 F/G 9/2 NL





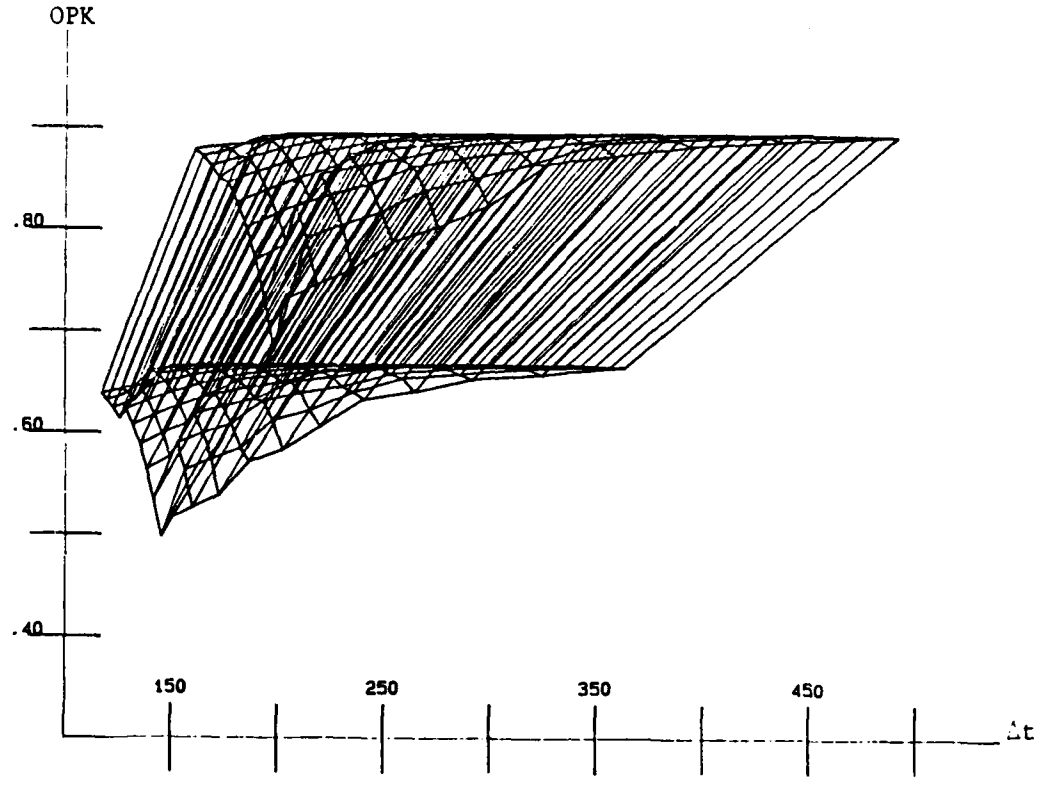


Figure 6.4 (c) Distance = 0 with half the uncertainty in δd

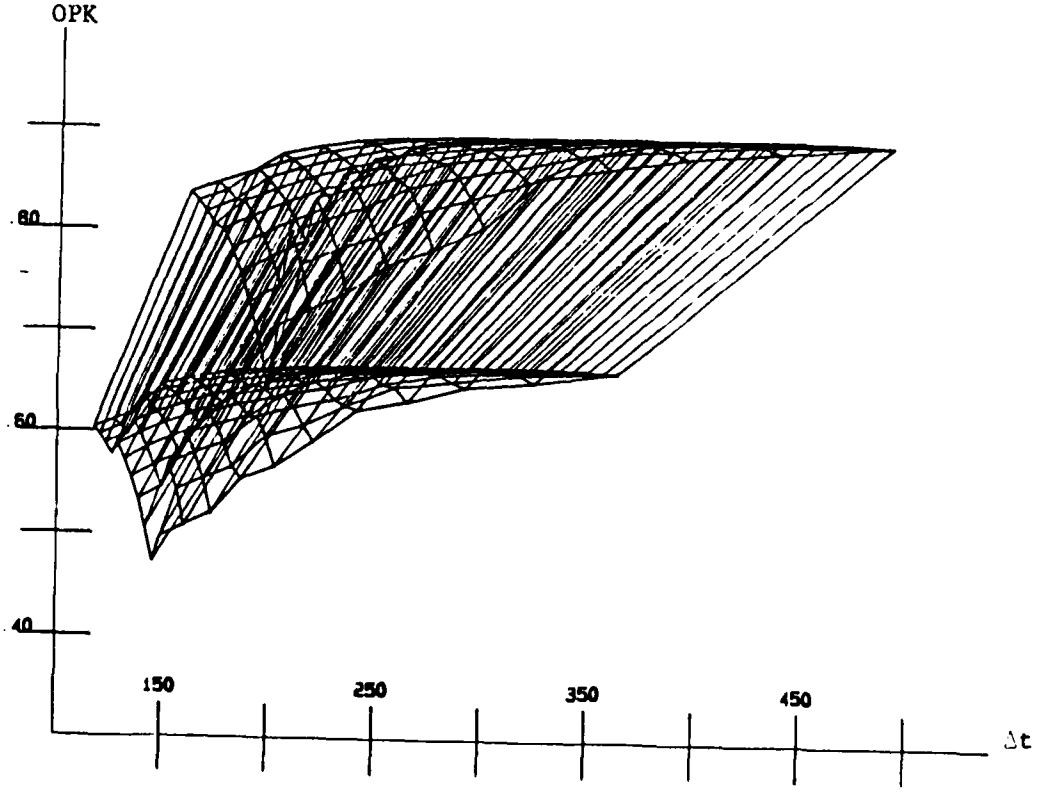


Figure 6.4 (d) Distance = 0 with the same uncertainties

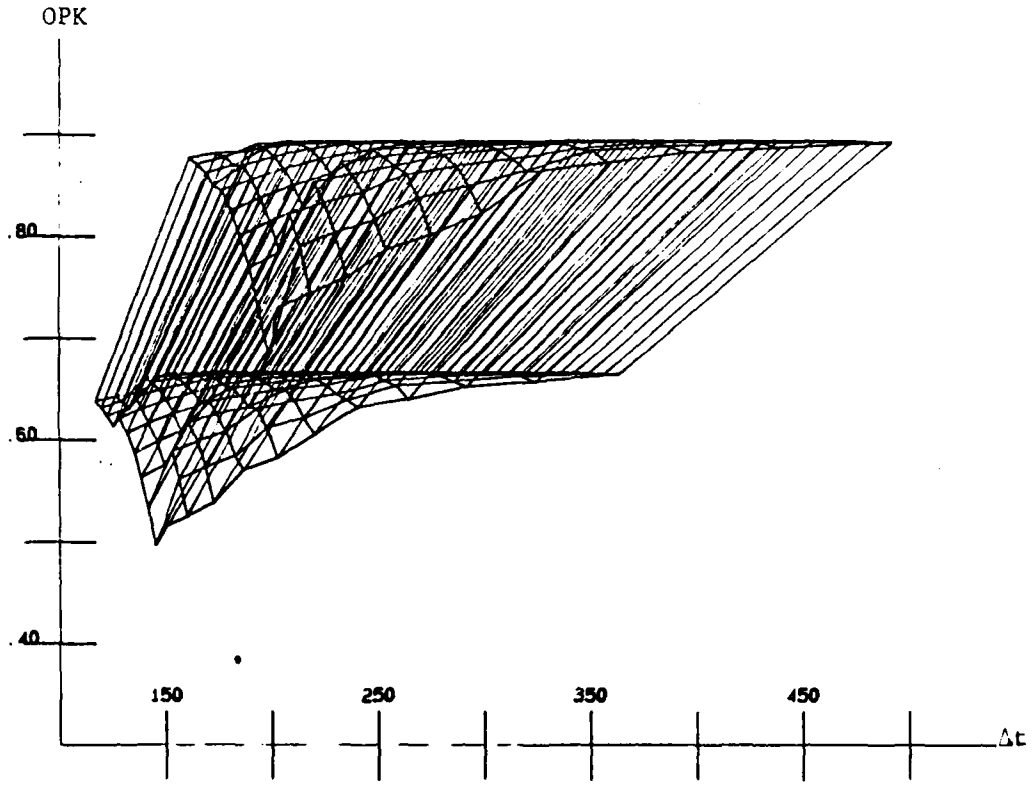


Figure 6.4 (e) Distance = 0 with twice the uncertainty in δd

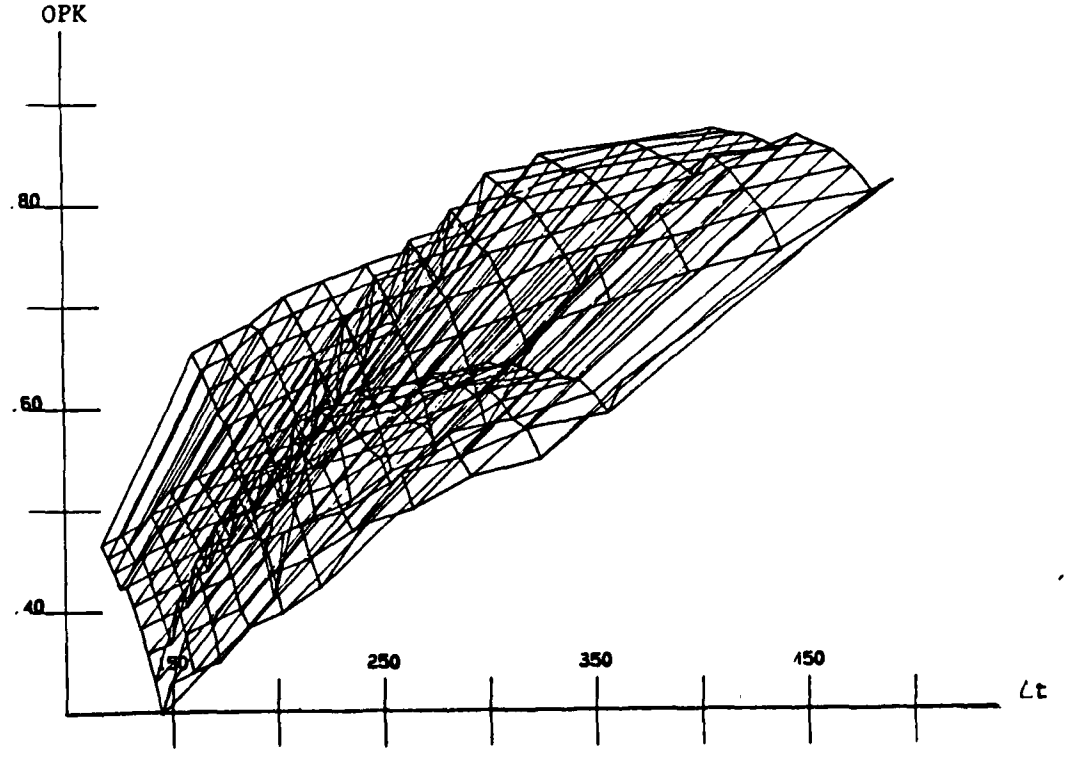


Figure 6.4 (f) Distance = 0 with twice the uncertainty in $\delta\beta$

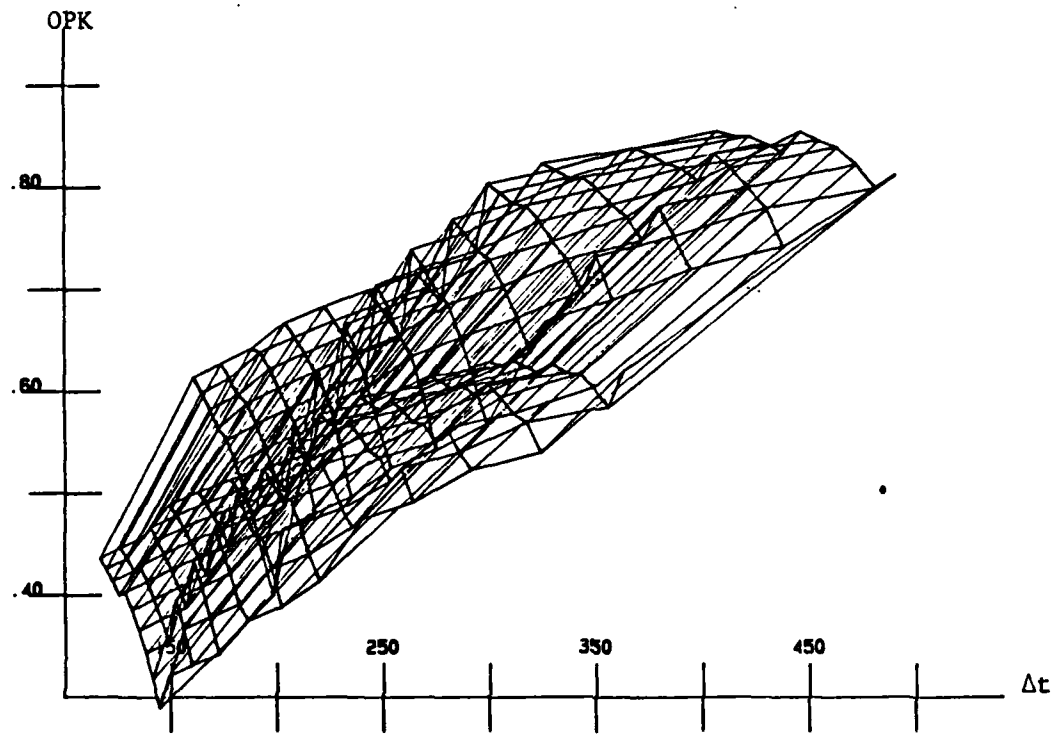


Figure 6.4 (g) Distance = 0 with twice the uncertainty in δd and $\delta\beta$

in the observation of each observer is the uncertainty in the estimate. These uncertainties can be fused into one uncertainty by taking the average of each one. Recall the relative uncertainty equation:

$$\rho=\delta\omega/\omega=\delta d/[(y_T-y_{FO})/2]\cdot(1/\sin\ \psi-1/\sin\ (\psi-\beta))+(\delta\beta/\ 2)\tan\ \beta/2 \qquad (6.1)$$

where

δd is the absolute error of the distance measurement equipment of the FO.

$\delta\beta$ is the absolute error in the reading of the angle β by the FO.

The absolute error in the timing equipment of the FO, δt, is considered equal to 0 for this problem. Thus, the only errors considered are δd and $\delta\beta$. If each FO has its own errors, then the errors are fused into one value by taking the average:

$$\delta d_{new} = (\delta d_1 + \delta d_2) / 2$$

$$\delta\beta_{new} = (\delta\beta_1 + \delta\beta_2) / 2 \qquad (6.2)$$

If $\delta d_1 = \delta d$ and $\delta\beta_1 = \delta\beta$, then let $\delta d_2 = \kappa_1 \cdot \delta d$ and $\delta\beta_2 = \kappa_2 \cdot \delta\beta$ where κ_i is some positive constant. Equation (6.2) then becomes

$$\delta d_{new} = ((\kappa_1+1) \cdot \delta d) / 2$$

$$\delta\beta_{new} = ((\kappa_2+1) \cdot \delta\beta) / 2 \qquad (6.3)$$

The series of loci in Figures 6.4(a) - 6.4(i) shows the results as the constants κ_1 and κ_2 vary. As the system loci show, if the second FO has a lower error of measurement, then the estimates are more precise and the overall OPKs increase; and if the second FO has a higher error of measurement, then the estimates are less precise and the overall OPKs

decrease. Note also, that the system locus is more sensitive to changes in the error of the angle, $\delta\beta$, than to changes in the error of the distance, δd. Changes in $\delta\beta$ result in drastic changes in the system loci. By looking at equation (6.1), the denominator of the term with δd in the numerator is relatively larger (on the order of 10) than the denominator of the term with $\delta\beta$ in the numerator, which is always less then 1.

6.3.2 The "Blind Spot" Case

Figures 6.5 show the system geometry of the case where a portion of the threat trajectory cannot be observed by either FO_1 or FO_2. This blind spot may be caused by a hill or some other object obstructing the view of FO_1.

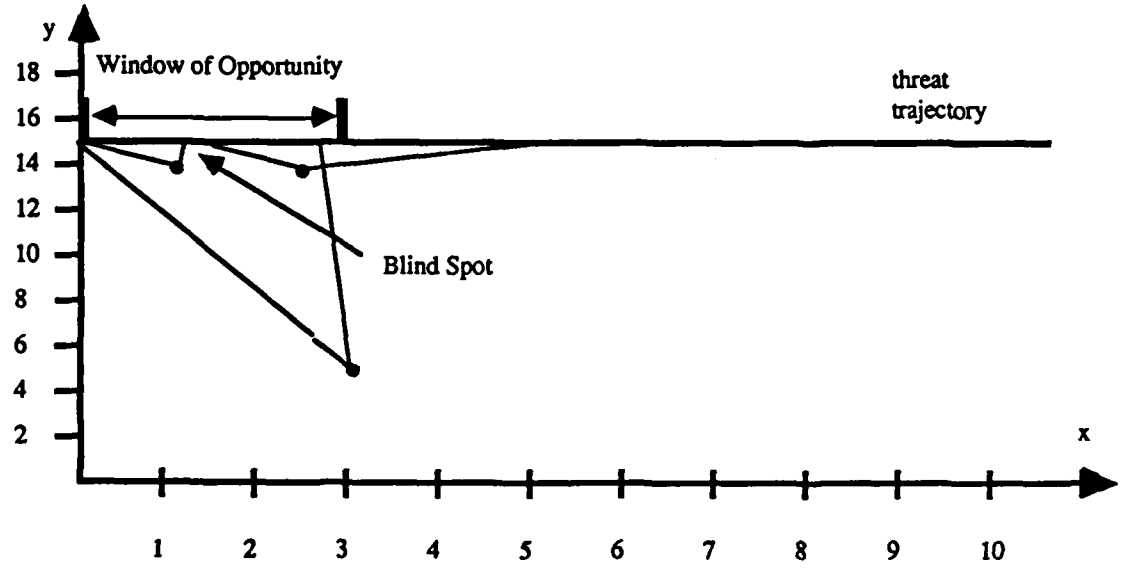


Figure 6.5 Geometry of a 'Blind Spot' on the threat trajectory

Thus, the minimum angle, ψ_{min} ,which constrains the observation has been increased to 85° as shown in Figure 6.5. This results in the 'blind spot' in which Doctrine 1 (L.S.L.S.) is applied for the time between the last obervation by FO_1 and the first observation by FO_2. The initial observation for Doctrine 1 is the last possible observation by FO_1.

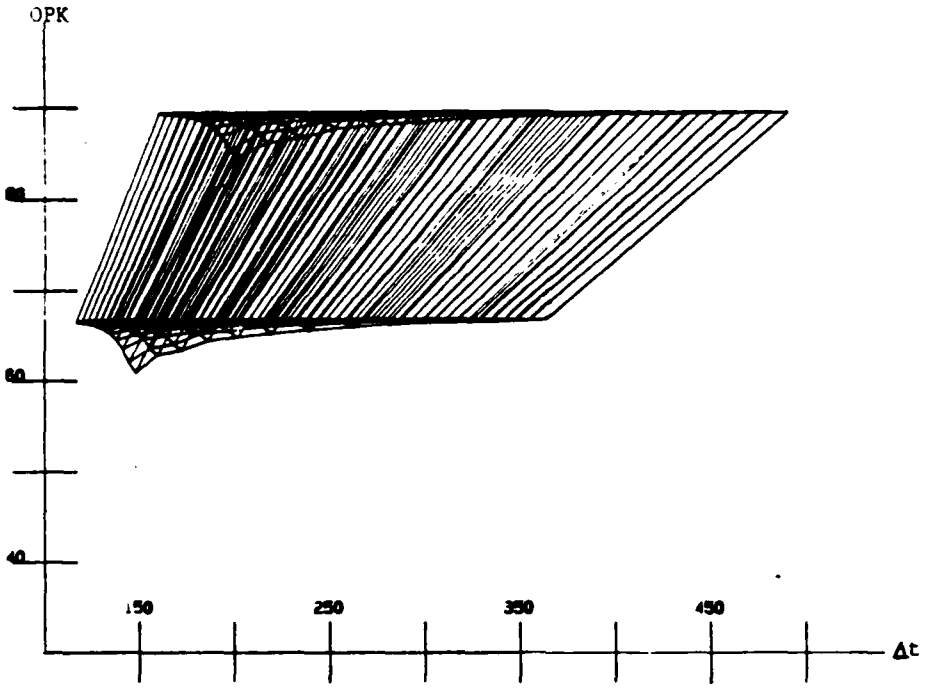


Figure 6.6 (a) Two FOs Separated by Distance d = 1.5 miles with blind spot

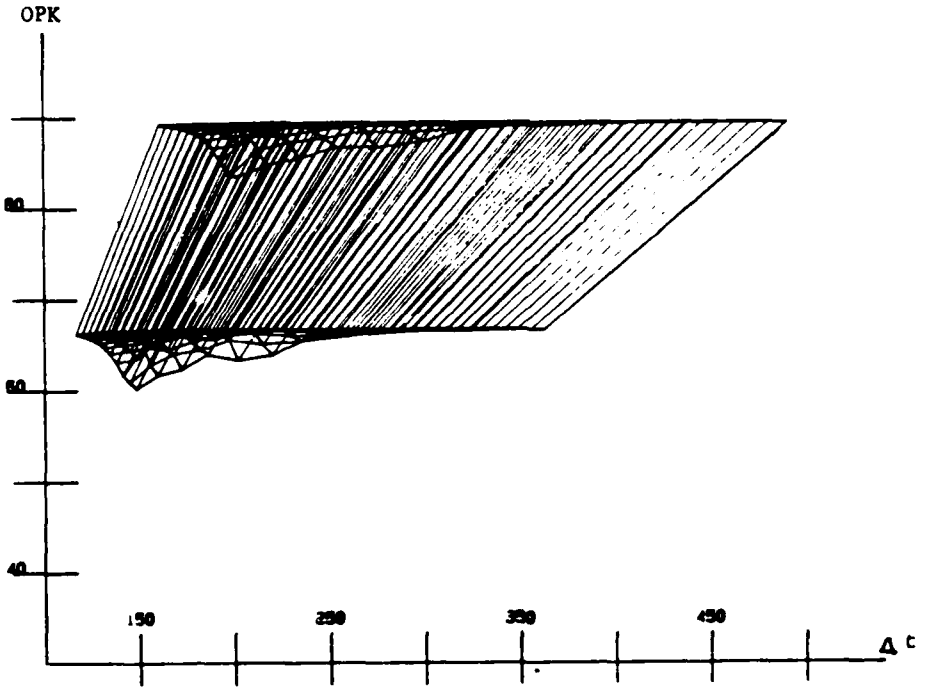


Figure 6.6 (b) Two FOs Separated by Distance d = 1.0 miles with blind spot

Figures 6.6(a) and 6.6(b) show the system loci. By having a blind spot in the system, the number of shots in each path increases substantially and often doubles. The more shots fired during this time, the higher the overall OPKs. Thus, as the distance between the FOs increases, as the system loci show, the overall OPKs increase.

6.4 The Two Battery Case: Two FOs under Doctrine 2 Separated by Distance 1.0 miles

The final case presented is the system with two FOs separated by distance d = 1.0 miles with two batteries. The geometry of the system is shown in Figure 6.7. Each option will be discussed.

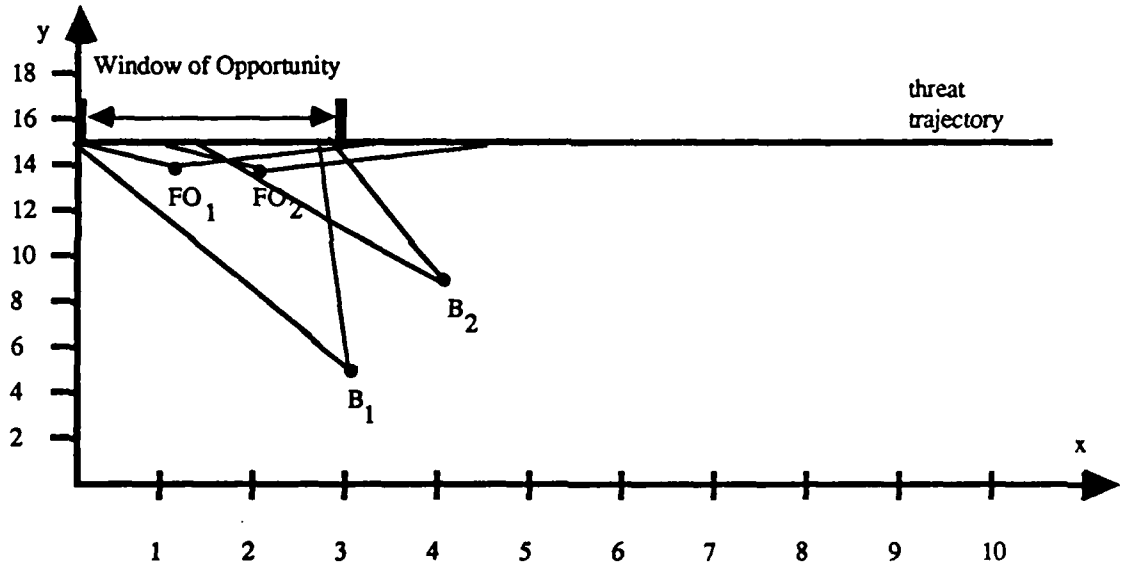


Figure 6.7 Two FOs in the Two Battery Case

Figure 6.8(a) shows the the system locus under option 1, the case of the uncoordinated batteries. The OPKs reach their saturation level at all values of β and there is no sensitivity to any changes in β. Figures 6.9(a) and 6.10(a) show orthographics projections of the three-dimensional loci of the one FO case and the two FO case respectively. The insensitivity to changes in β of the two FO case is futher displayed. This insensitivity makes

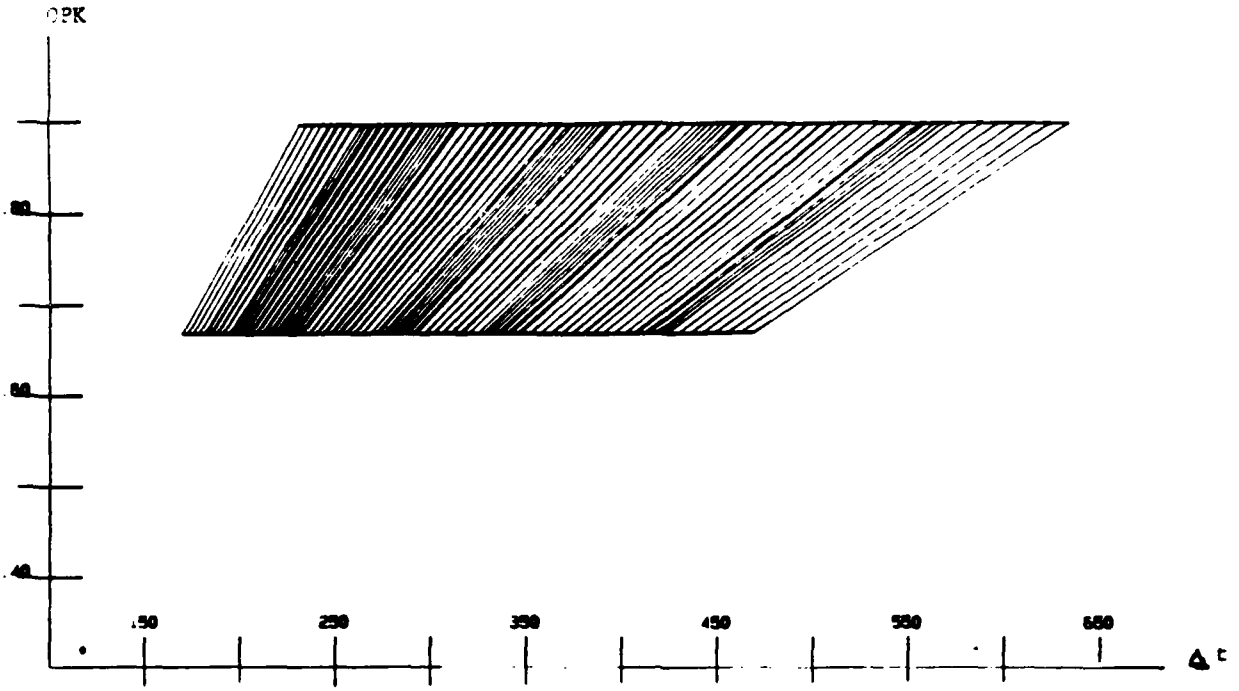


Figure 6.8 (a) Two FOs Separated by Distance d = 1.0 miles under Option 1

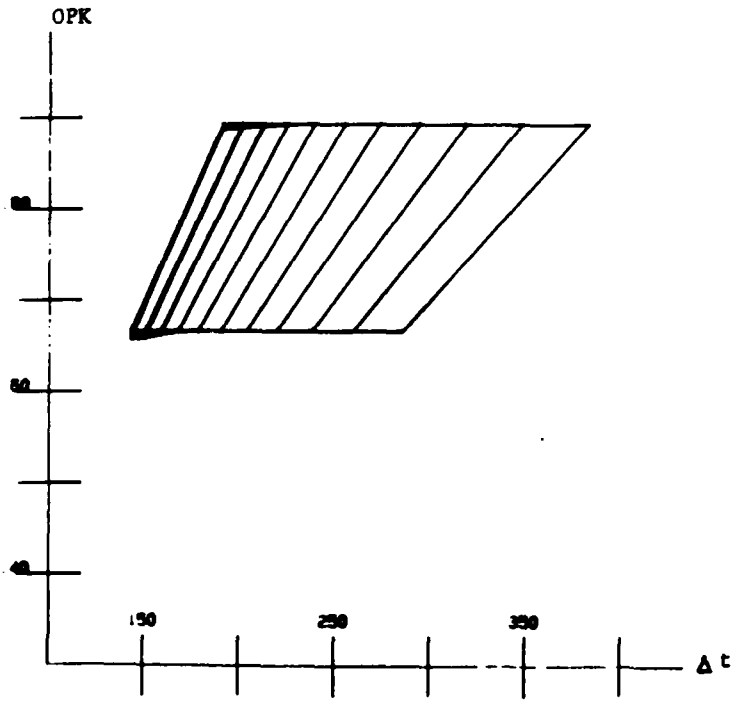


Figure 6.8 (b) Two FOs Separated by Distance d = 1.0 miles under Option 2

Locus is rotated 25° around the
x axis, 25° around the y axis,
and -25° around z axis.

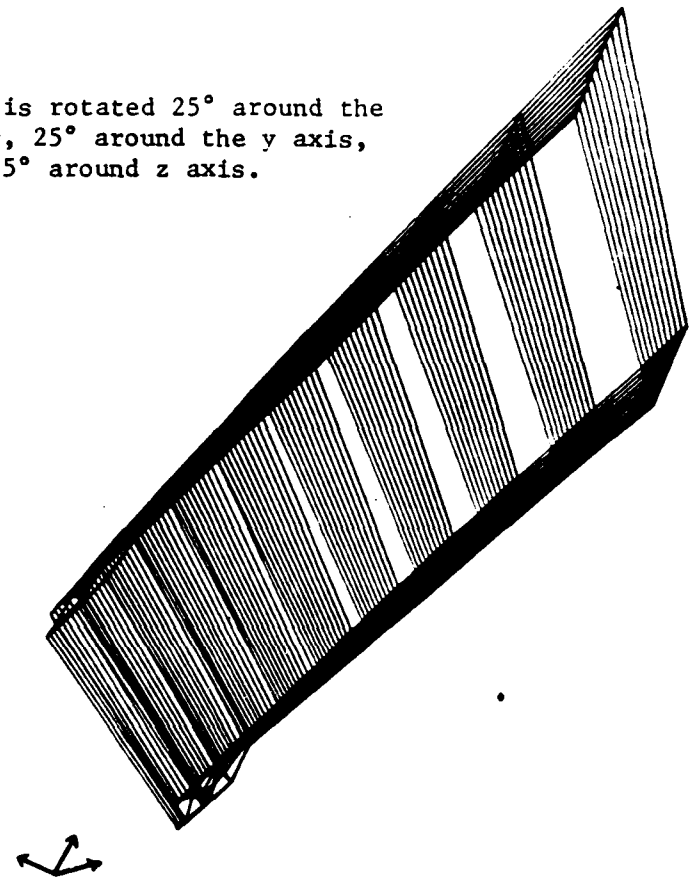
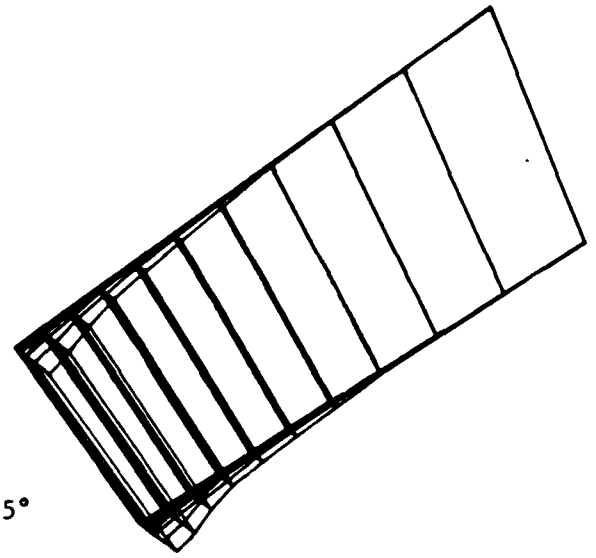


Figure 6.9 (a) Doctrine 2, Option 1 with one FO.



Locus is rotated 25°
around the x axis,
25° around the y axis,
and -25° around the z axis.

Figure 6.9 (b) Doctrine 2, Option 2 with one FO.

Locus is rotated 25° around the x axis, 25° around the y axis, and -25° around the z axis.

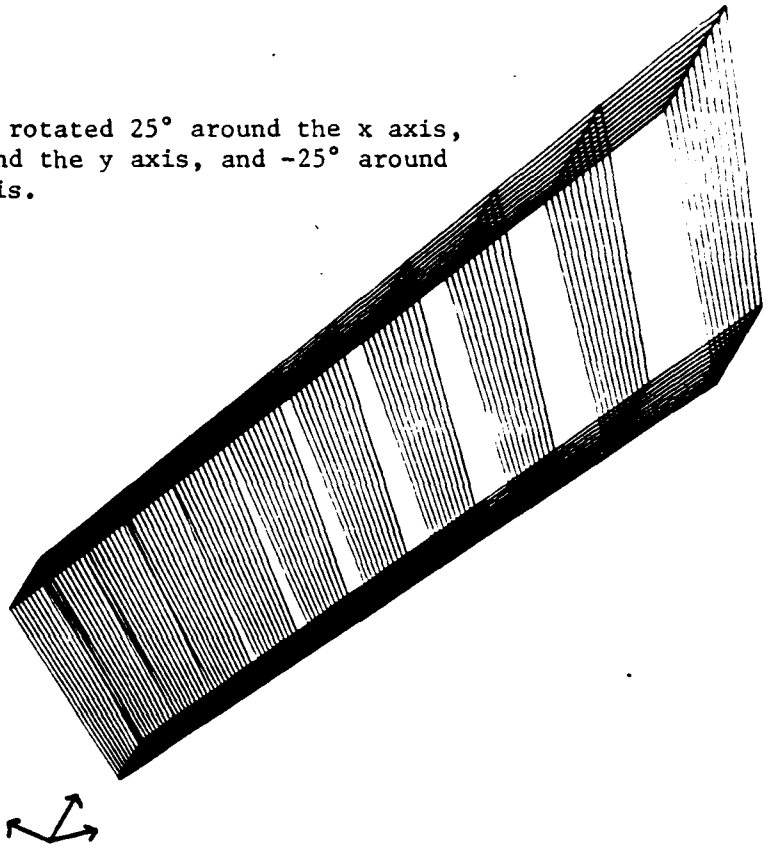
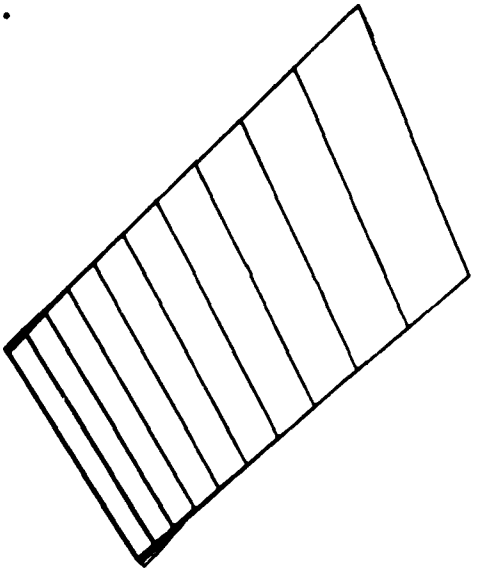


Figure 6.10 (a) Doctrine 2, Option 1 with 2 FOs.



Locus is rotated 25° around the x axis, 25° around the y axis, and -25° around the z axis.

Figure 6.10 (b) Doctrine 2, Option 2 with 2 FOs.

sense since the batteries are shooting whenever they posssibly can and are firing on continuously updated estimates of the threat position.

Figure 6.8(b) shows the system locus under option 2, the case of coordinated batteries. There is a small sensitivity to β, but only at high speeds. This sensitivity can further be seen by comparing the rotated Figures 6.9(b) and 6.10(b). Although some of the OPKs are smaller than those of Option 1, the same OPKs can be achieved with Option 2 at subtantially less cost because fewer shots are fired.

6.5 Conclusion

This analysis shows that a second FO does not make the system significantly more effective. The same OPKs can be achieved as in the one observer case. To further improve the system, the probabilities associated with the links must be improved. Adding a second observer will cause the system to acheive this saturation point but at an expense. Nonetheless, other interesting properties of the system have been found with the use of the graphics system and models derived from systems effectiveness analysis. One interesting feature in the sensitivity analysis to the changes in the error of $\delta\beta$ and δd. This type of analysis can easily be carried out with the use of the graphics system.

CHAPTER VII

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

7.1 Conclusions

This thesis has demonstrated the use of a computer graphics system in system effectiveness analysis. By studying the loci with different projections, a more thorough analysis can be completed. Not only will measures of effectiveness be computed, but other properties of the loci can be studied as well.

For example, for the TACFIRE case studied in this thesis, many loci were plotted, which was needed to understand the effect of adding a second observer to the system. As the research progressed, other variations of the problem were explored such as the sensitivity to errors in the measuring equipment of the system. Also, other constraints were exchanged, such as the areas of observation of FO_1 which caused the blind spot. Once the data was generated for these different systems, the plots could be viewed in a matter of seconds. The concept of using computer graphics in science and engineering, although not new, is just beginning to realize its potential with dedicated workstations such as the system described in this thesis.

7.2 Suggestions for Future Research

The graphics program for the engineering workstation is the beginning of a series of blocks that will be built to make the system more useful. One of these blocks is incorporating the computation of the measures of effectiveness using the loci. One measure of effectiveness is the computation of the volume intersection of the mission and system loci. If this were integrated into the existing software, a user could quickly determine one measure of effectiveness and thus the analysis would be enhanced by this information. By studying the loci with the graphics system, more measures of effectiveness will be determined and incorporated into the workstation.

One aspect of the program that could be improved is the user interface, which is presently a simple menu system (see Appendix C, The User's Manual). Developing a user-interface is quite difficult and time-consuming, but as the workstation develops, it will

have to be taken into consideration.

A final subject that should be researched is the sensitivity of system effectiveness to measurement errors in the system. In Chapter 6, it was found that the TACFIRE system was relative by more sensitive to changes in the error of the observation angle $\delta\beta$ than the distance measurement δd. Although briefly studied here, some interesting research may result from this. Note that no specific reccommendations can be made on the reliability/survivability of the two FO system since a network of links and nodes was not rebuilt and analyzed for this case.

REFERENCES

Barlow, R.E., and F. Proschan, (1974), Statistical Testing or Reliability and Life Testing, Holt, Rinehart, and Winston, New York

Bouthonnier, Vincent, (1982), "Systems Effectiveness Analysis for Command and Control", LIDS-TH-1231, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139, August 1982.

Chapra, Steven C., and Raymond P. Canale, (1985), Numerical Methods for Engineers with Personal Computer Applications, New York: McGraw Hill Book Co.

Cothier, Philippe H., (1984), "Assessment of Timeliness in Command and Control", LIDS-TH-1391, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139, August 1984.

Karam, Joseph Ghaleb, (1985), "Effectiveness Analysis of Evolving Systems", LIDS-TH-1431, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139, January 1985.

Levis, Alexander H., Paul K. Houpt, and Stamatios K. Andreadakis, (1984), "Effectiveness Analysis of Automotive Systems", LIDS-P-1383, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139, June 1984.

Newman, William M. and R. F. Sproull, (1979), Principles of Interactive Computer Graphics, New York: McGraw Hill Book Co.

Tufte, Edward A., (1983), The Visual Display of Quantitative Information, Chesire, Conn.: Graphics Press.

Washington, Lisa Anne, (1985) "Effectiveness Analysis of Flexible Manufacturing Systems", LIDS-TH-1430, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 02139, January 1985.

IBM Personal Computer Professional Graphics Controller Technical Reference,(1984), IBM Personal Computer Hardware Reference Library, August 5, 1984.

Appendix A

The workstation used for the effectiveness analysis of C^3 systems consists of:

Hardware Specifications:

• IBM-PC/AT with 512K of Random-Access Memory (RAM)	Product # 5170-099	
• IBM Professional Graphics Controller	Product # 6451-501	
• IBM Professional Graphics Display	Product # 5175-001	
• Hewlett-Packard 6-pen Color Plotter	7475A Plotter	RS232-C /CCITT v.24
• Alloy Computer Tape Backup System	Product Name: FT-60	
• Epson Dot-Matrix Printer	Model: FX/100	

Software Tools -- Specifications:

• IBM Graphics Development Toolkit	v. 1.0	Product # 6024-196
• IBM Professional Fortran Compiler	v. 1.0	Product # 6024-200
• MicroSoft MacroAssembler	v. 2.0	Part # 016-014-011
• TURBO Pascal Compiler	v. 3.0	Borland, Inc.
• MS-DOS	v. 3.0	Microsoft, Inc.

Appendix B

ASSEMBLY LANGUAGE ROUTINES

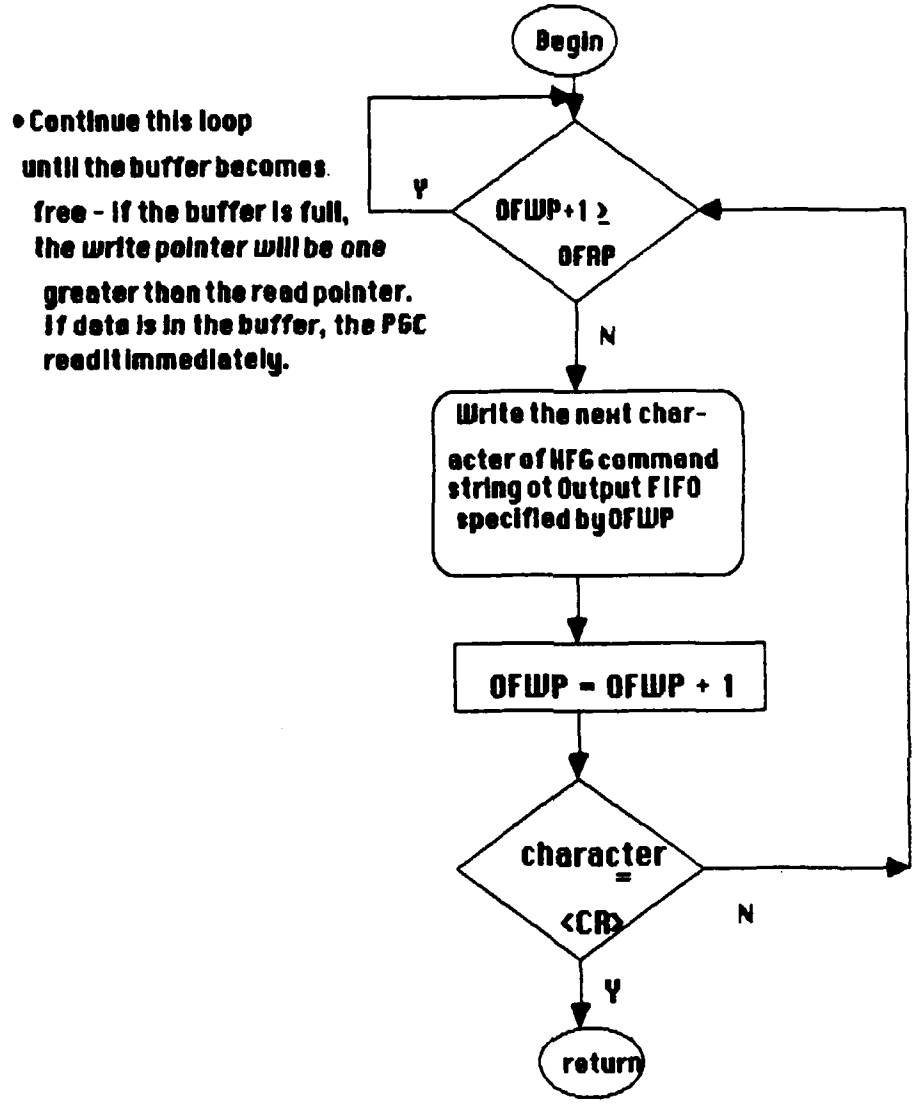
Flowchart for COM

- This program sends high function graphics commands (ASCII character strings) to the Professional Graphics Controller which will then interpret the commands and send data to the screen.

- Refer to the Professional Fortran - Installation and Use, Appendix C, for the conventions used for calling assembly language routines in FORTRAN and the IBM Professional Graphics Controller Technical Reference Manual , pp. 78-82, for specifications on communicating with the controller.

- The high function graphics commands are in the form of ASCII character string are stored in parameter block by FORTRAN (see Appendix C).

- Pointers to buffers are incremented by one modulo 255 --- addition is in hexadecimal: if OFWP=FF, OFWP+1=FF+1=0 because the carry bit is lost.



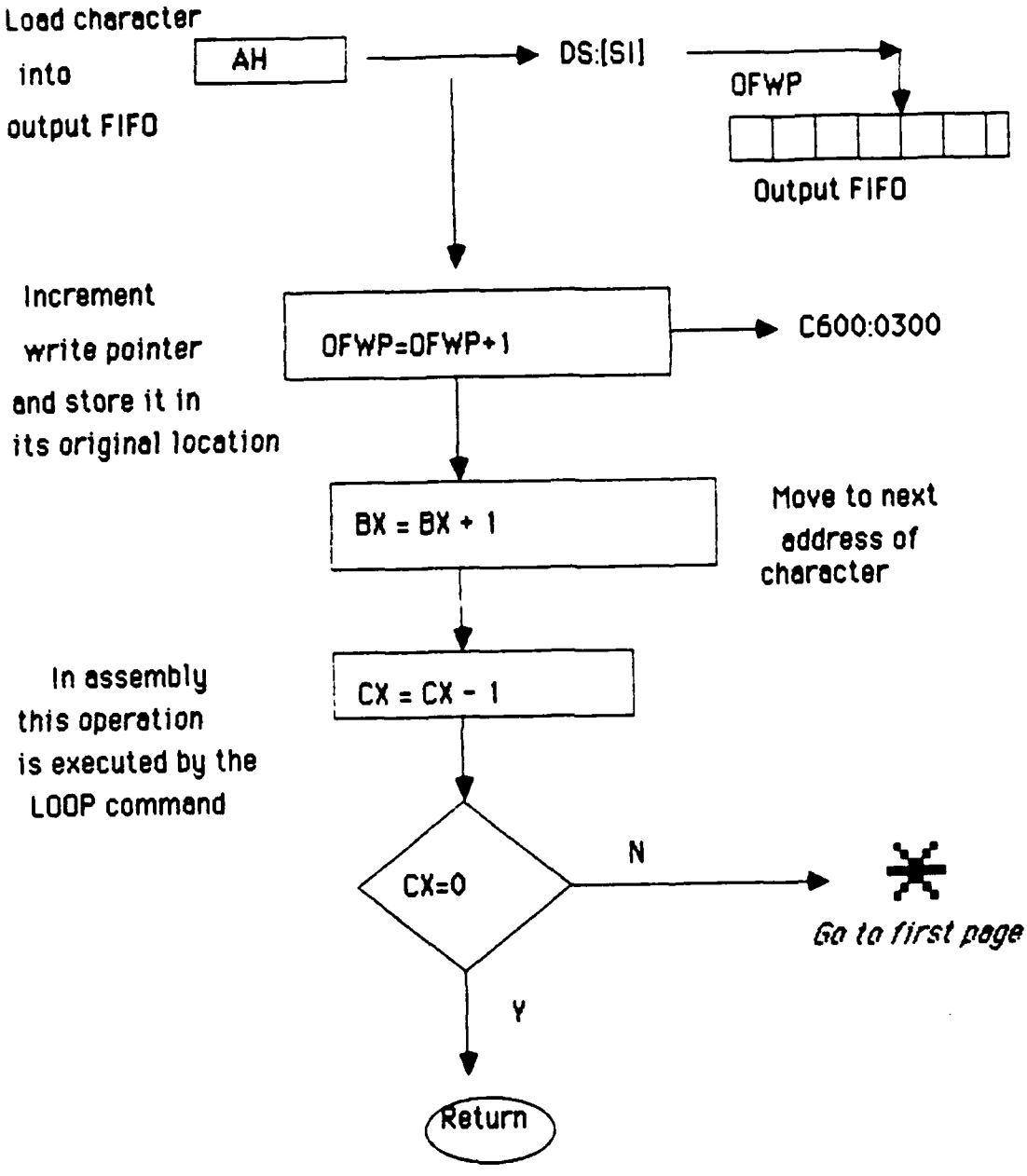
KEY WORDS:

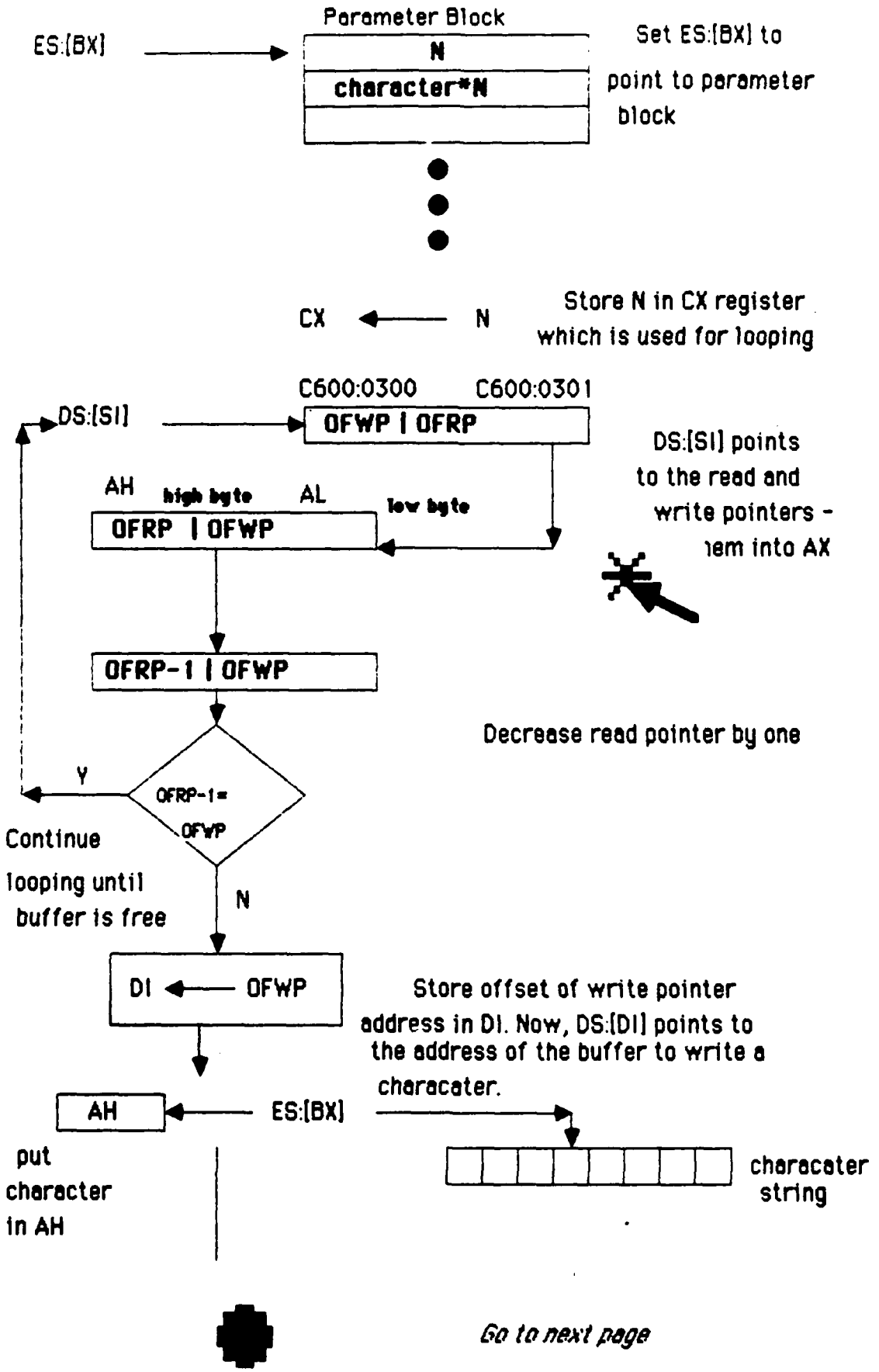
PGC: Professional Graphics Controller- the graphics card in the IBM-PC/AT

OFRP: Output FIFO (First In First Out buffer) Read Pointer

OFWP: Output FIFO Write Pointer

HFG: High Function Graphics


Load character
into
output FIFO
AH
DS:[SI]
OFWP
Output FIFO
Increment
write pointer
and store it in
its original location
OFWP=OFWP+1
C600:0300
BX = BX + 1
Move to next
address of
character
In assembly
this operation
is executed by the
LOOP command
CX = CX - 1
CX=0
N
Go to first page
Y
Return



Go to next page

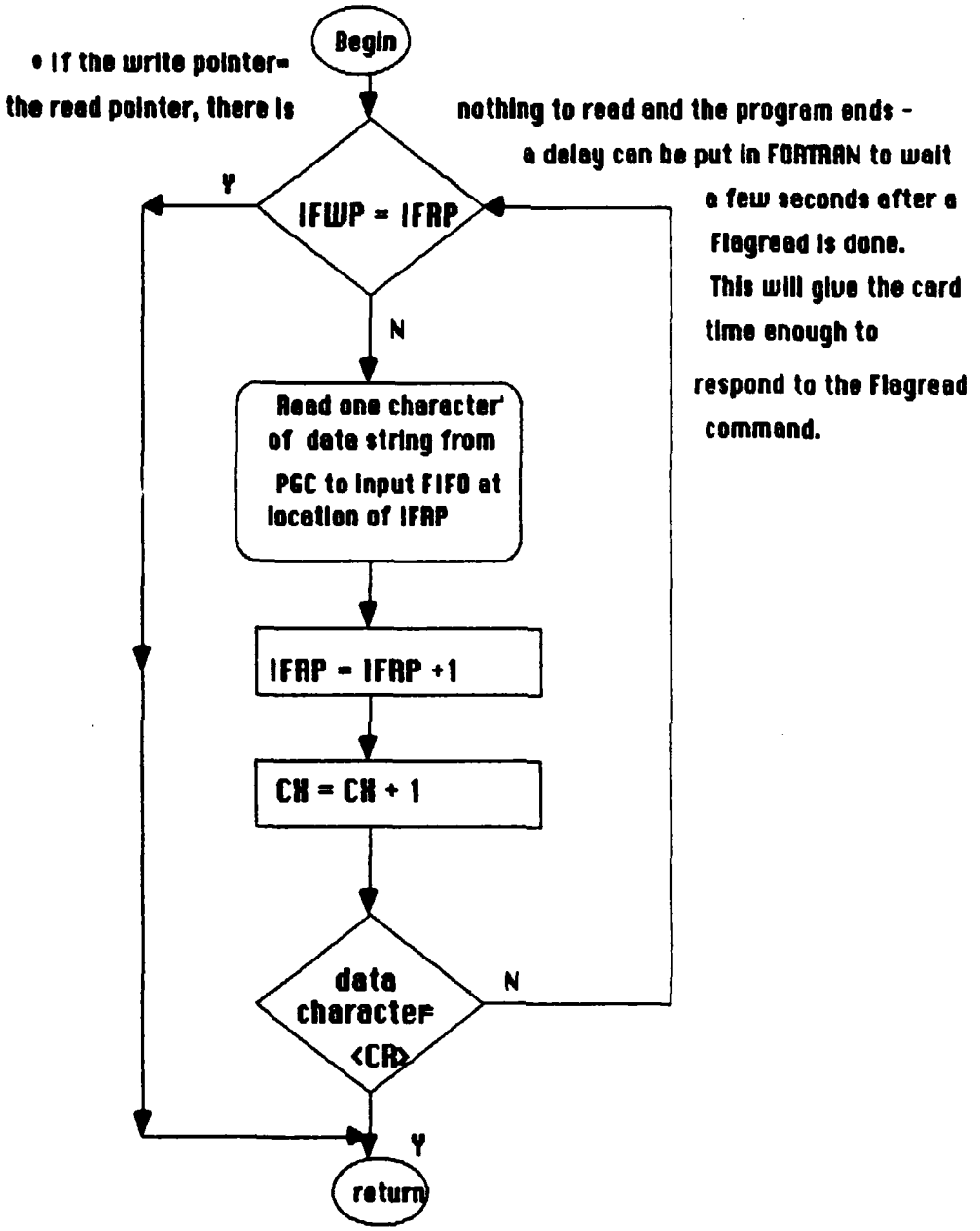
Flow chart for INFIFO and ERFIFO

- This program reads data and messages from the PGC as character strings.

- Refer to Professional Fortran - Installation and Use, Appendix C for the conventions used for calling assembly language routines in FORTRAN and to the IBM Professional Graphics Controller Technical Reference Manual, pp. 78- 82.

- The CX register starts with value 0. At the end of the routines, it will hold the length of the character string.

  Note : Although this example shows reading from the input FIFO, the algorithm is exactly the same for reading from the error FIFO, except that the offset to the buffer is $C600:0305



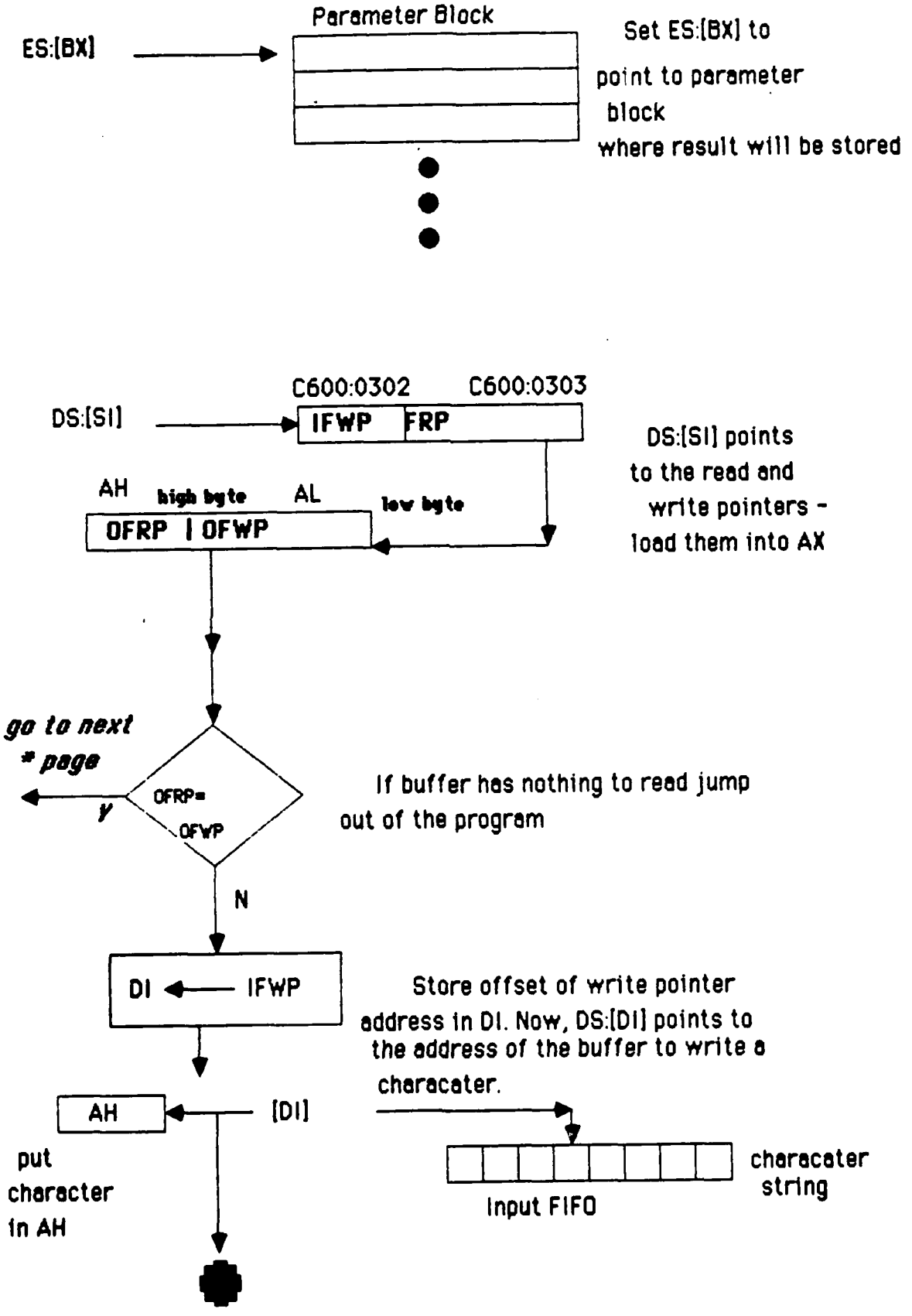
KEY WORDS:

PGC: Professional Graphics Controller- the graphics card in the IBM-PC/AT

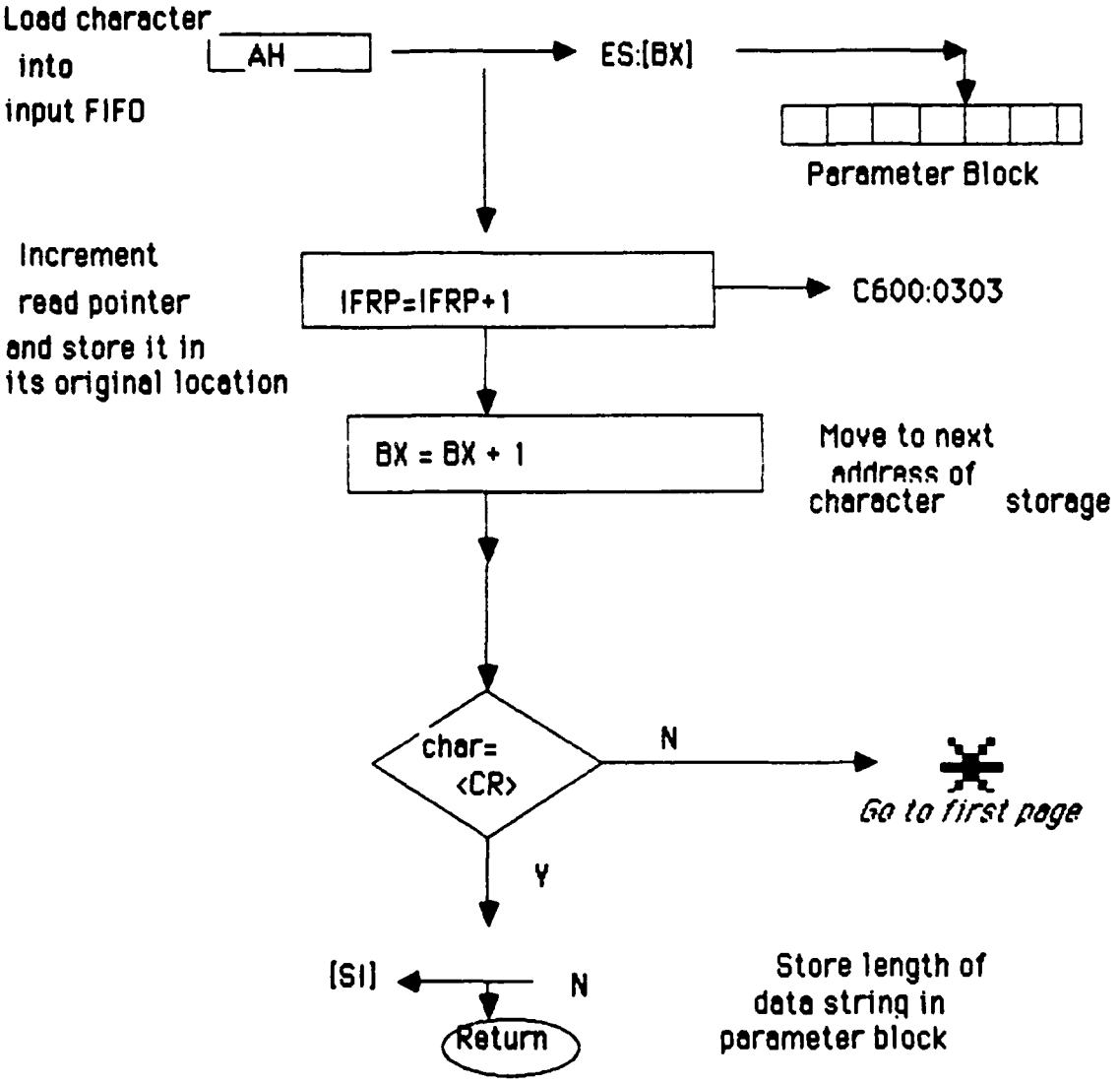
IFRP: Input FIFO (First In First Out buffer) Read Pointer

IFWP: Input FIFO Write Pointer

Detailed Flowchart of Assembly Language Program-INFIFO



Go to next page



NOTE: This flowchart is the same for ERFIFO expect the pointer and buffer reside in different locations then above.

Appendix C

User's Manual for Graphics Software

Locus-Construction Program

User's Manual

by
Christine M. Bohner
Massachusetts Institute of Technology
Laboratory for Information and Decision Systems
Spring 1986

I. Introduction

This program is designed to plot data obtained by using System Effectiveness Analysis (SEA). Technical details on the software can be obtained from Bohner (1986), "Computer Graphics for System Effectiveness Analysis" and an annotated listings of the source code can be obtainted from the Program Diskette. With use of the program, data can be viewed in different ways. To run, the program needs an IBM-PC/AT computer with a Professional Graphics Controller card, a Professional Graphics Display, MS-DOS version 3.00, and 512K of random-access memory (RAM). The code was written in Professional FORTRAN and Microsoft Macro-Assembly Language, but if you have the executable file, you do not need the compiler to run the program.

II. Data Files

None of the data plotted is generated by the graphics program. The data is read from a formatted data file. Each line of data must have the following format:

line 1: imax jmax kmax lmax

These are the maximum indices over which the data may vary. If an index is not used, its maximum should be assigned to 1 and **not** 0.

line 2 - end of file: I J K L x_1 x_2 x_3 x_4 x_5 x_6 x_7

Number entries must have at least one space between each other.

I, J, K, and L are the index values and presently have no use in the program.

x_1,....,x_7 are the data points in space (up to 7 dimensions can be assigned).

Please refer to Bohner (1986) and source code for further details on the data structure used in the program.

III. Menus

When you are ready to run the program, type

C > locus

A series of menus will appear. They will be explained in sequence here.

Title Menu

```
========================================
||                                    ||
||         LOCI  CONSTRUCTION         ||
||              PROGRAM               ||
||                                    ||
||    by Christine Bohner             ||
||    Laboratory for Information      ||
||    and Decision Systems            ||
||    Spring 1986                     ||
||                                    ||
========================================
Press space bar and return to continue...
```

Simply follow the directions...press the space bar and return.

The following menus have options with defaults in parenthesis beside them. If the option is a yes or a no choice, selecting it will act like a switch; if the default answer is a 'no', and you select the option, the menu will next appear with the default as a 'yes'. Other options will prompt you with another small menu of sub-options. Once the menu is set with the defaults you want, you can choose the 'go ahead and plot' option - option 6.

Data Menu

```
DEFAULT VALUES ARE IN PARENTHESES BESIDE MENU OPTIONS.

==================================================
|| ******* DATA MENU ********                    ||
||   1. Read file(Y)                             ||
||   2. Read index max. from file(Y)             ||
||   3. Set max. # of points to read(N)          ||
||   4. Set output device (CRT       )           ||
||   5. Set order of indices (N)                 ||
||        (LIJK)(LIKJ)(LKJI)                     ||
||   6. Plot Data                                ||
||   7. Quit                                     ||
==================================================

     Menu Option>>>>>6

              Data File Name>>>>doc1.dat
```

- Option 1: This allows you to plot different files if you have already been running the program for a session. If you have just begun the session, and you do not specify a file, the program will not continue until you do so.
- Option 2: If the file you are reading is missing line 1 (the maximum number of indices as specified earlier), the default will be assigned to imax, jmax,kmax, and lmax -- 11,11,11, and 1 respectively.
- Option 3: The maximum number of points can be determined by the program (max. # of points = imax * jmax * kmax * lmax), but if you want to specify the number yourself, choose this option. If you specify too many points, the program will tell you that you have read beyond the end of the file.
- Option 4: This is where you can direct the graphics output to the CRT, the Printer, or the

Plotter. Note, however, that 3-D viewing can only be viewed on the screen. The default is the CRT.

- Option 5: The order of the indices specifies how the data will vary which will allow different viewing. The default, seen on the menu, can be changed here to any order desired.
- Option 6: When all the defaults are how you want them, select this option.
- Option 7: QUIT!

```
DEFAULT VALUES ARE IN PARENTHESES BESIDES MENU OPTIONS.
```

Plot Menu

```
==========================================================
||              **** PLOT MENU ****                      ||
||1. Set Dimensions(          1,          2,          3)||
||2. Set Graphics as (P)oints or (L)ines (L)             ||
||3. Display in 3D (Y)                                   ||
||4. Supress axes drawn in 2-D     (N)                   ||
||5. Set scale factors (sx,sy,sz)  (N)                   ||
||6. Start Plot                                          ||
||7. Return to previous menu                             ||
==========================================================

              Menu Option>>>>>3
```

- Option 1: Only three dimensions can be viewed on the screen at one time. The default is dimensions are 1, 2, and 3, but any three dimensions 1 through 7 can be chosen in any order.
- Option 2: Although the default is lines, you may also view the data as points on the screen.
- Option 3: If your do not choose 2-D viewing, the view-mode will automatically go to 3-D.
- Option 4: If you are viewing in 2-D and do not want the axes displayed, select this option. This is useful if you want to plot rotated axes.
- Option 5: This allows you to set your own scale factors for each axis you are plotting If certain dimensions of your data are in the same units then you want to set these dimension on the same scale. This option is also useful if you want to enlarge or reduce the picture you are plotting. See Figure 1 for an example of what occurs in the

program when this option is selected.

- Option 6: When all the defaults are as you like, select this option.
- Option 7: If you would like to return to the previous menu above(to read another file, perhaps), you can choose this option.

```
          Menu Option>>>>>6

****** MAPPING STATISTICS  **********
**     Dimension            1        **
**     Maximum =      490.40109253   **
**     Minimum =      116.65639496   **
**     Scale   =       43.83607483   **
****************************************
****** MAPPING STATISTICS  **********
**     Dimension            2        **
**     Maximum =        0.89389998   **
**     Minimum =        0.54470003   **
**     Scale   =    46917.24609375   **
****************************************
****** MAPPING STATISTICS  **********
**     Dimension            3        **
**     Maximum =      701.62481689   **
**     Minimum =      350.81240845   **
**     Scale   =       46.70159912   **
****************************************

** Old scale factors: SX =       43.83607483
                      SY =    46917.24609375
                      SZ =       46.70159912
New factors: SX, SY, SZ (number = #.XXXX) >>>>50.0000 24000.0000 50.0000
```

Figure 1

3-D Menu

```
DEFAULT VALUES ARE IN PARENTHESES BESIDE MENU OPTIONS.

==========================================================
||  ****  3D MENU  ****                                   ||
||  1.  Set rotation in x (0     )                       ||
||  2.  Set rotation in y (0     )                       ||
||  3.  Set rotation in z (0     )                       ||
||  4.  Set translation (0 0 0                  )        ||
||  5.  Save transformed data (N)                        ||
||  6.  Start 3D plot                                    ||
||  7.  Return to previous menu                          ||
==========================================================

   Menu Option>>>>>6
```

•Option 1: Rotation around the x-axis using the right-hand rule. The point of rotation is the center of the locus and rotation should be specified in degrees.

• Option 2: Rotation around the y-axis.

• Option 3: Rotation around the x-axis.

• Option 4: Translation of the loci on any of the three axis. Note that translation along the z-axis which is pointing out of the screen will not be seen because the projection is orthographic onto the screen.

• Option 5: When you want to save the transformed data to a file for printing or plotting, choose this option. ***NOTE***: Sometimes, especially when the program saves large files to the hard disk, there is a write error; your data will not look as expected when plotted. Try to save the data again.

• Option 6: When all the defaults are as you like, select this option.

• Option 7: If you want to return to the previous menu, choose this option.

This manual is just for use of the program....if you need to alter the source code or other technical details, please refer to the source code and Bohner (1986).

Appendix D
Derivations of Equations

This appendix will present the derivations of equations mentioned in Chapter 4 and will be split into different sections, as follows:

- D.1 Analysis of the Time of Flight $\Delta\tau_3$
- D.2. Derivation of the Uncertainty Expression $\rho=\delta\omega/\omega$
- D.3. Network Analysis
- D.4. Derivation of the System Attributes for the Two Battery Case

Most of the derivations here closely follow the development in Cothier's (1984) Appendix A through Appendix E; thus, the presentation here will be brief.

D.1 Analysis of the Time of Flight $\Delta\tau_3$

Assume that the projectile follows Newtonian physics without air resistance. If a projectile is given an initial velocity V_o then the motion is depicted in Figure D.1.

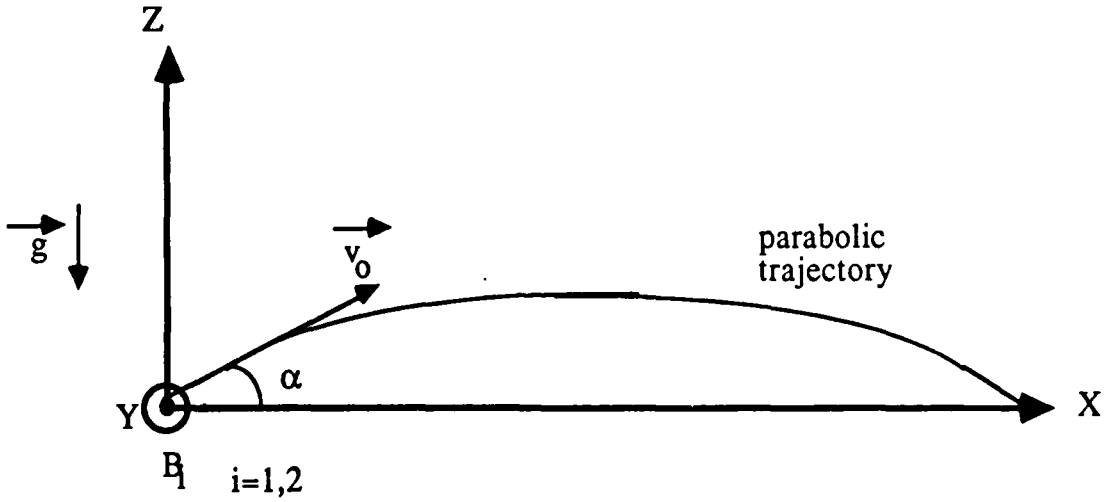


Figure D.1. The Projectile Trajectory

From Newtonian mechanics, the equations of motion are:

$$X = V_0 \cos \alpha \, (t - t_0) \qquad \text{(D.1)}$$

$$Y = 0 \qquad \text{(D.2)}$$

$$Z = -1/2 \, g \, (t - t_0)^2 + V_0 \sin \alpha \, (t - t_0) \qquad \text{(D.3)}$$

The time the battery fires is t_0 with an angle to the horizontal plane of α. Solving for V_0 in equation (D.1) and substituting into equation (D.3) yields

$$Z = -\frac{g}{2 V_0^2 \cos^2 \alpha} X^2 + X \tan\alpha \qquad \text{(D.4)}$$

From Figure D.1, it is shown that the projectile hits the ground at Z=0. Therefore, equation D.4 becomes (using the trigonometric identity $\sin 2\alpha = 2 \cos \alpha \sin \alpha$)

$$X = \frac{V_0^2}{g} \sin 2\alpha \qquad \text{(D.5)}$$

The maximum range the projectile can reach is when the angle $2\alpha = 90°$ or $\alpha = 45°$. Let

$$R = \frac{V_0^2}{g} \qquad \text{(D.6)}$$

The time delay between the time the battery fires and the time the projectile hits the ground is $\Delta\tau_3$, which is

$$\Delta\tau_3 = t_{impact} - t_o \qquad (D.7)$$

The range X of the projectile can be determined by equations (D.1), (D.5) and (D.7):

$$X = V_o \cos\alpha \cdot \Delta\tau_3 = (V_o^2/g) \sin 2\alpha$$

Solving for time yields:

$$\Delta\tau_3 = (2\ V_o/g) \sin\alpha$$

And finally substituting in with equations (D.6)

$$\Delta\tau_3 = 2\sqrt{\frac{R}{g}} \sin\alpha \qquad (D.8)$$

Now X, the range of the projectile can also be expressed in terms of the geometry of Figure 4.1. This geometry is summarized in Figure D.2:

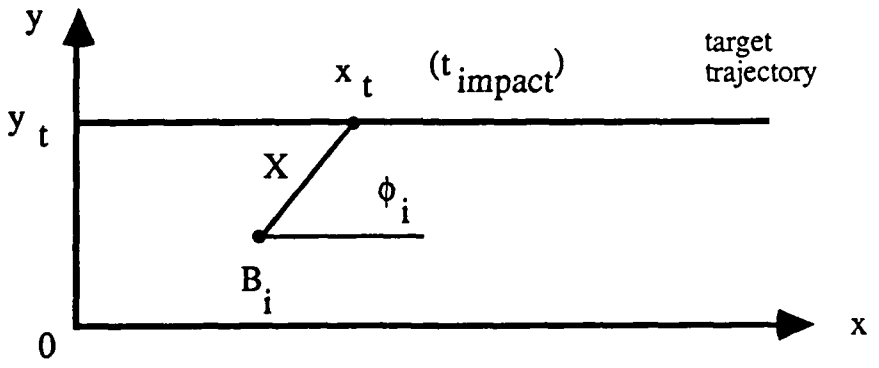


Figure D.2. Geometry of the Shooting of the Projectile

$$X = \frac{y_t - y_{B1}}{\sin \phi_i} \qquad i = 1, 2 \qquad (D.9)$$

Note also, that from equations (A.5) and (A.6)

$$X = R \sin 2\alpha \qquad (D.10)$$

Setting equations (D.9) and (D.10) equal:

$$\sin 2\alpha = (y_t - y_{B1}) / [R \sin \phi_i] \qquad i = 1,2 \qquad (D.11)$$

Numerical Values

The numerical values of $\Delta\tau_3$ are different for batteries B_1 and B_2 because their distances to the target trajectory differ. From Figure 4.1 (which shows the geometry of the system), the minimum time delay for both batteries (and thus, the shortest range X) corresponds to the minimum angle ϕi. To determine this delay, the following equation is used:

$$\sin 2\alpha = [y_t - y_{Bi}] / [R \sin \phi_{i\ min}]$$

After solving for α , $\Delta\tau_3$ can be determined by equation (D.8).

The maximum time delay corresponds to the earliest firing of the projectile. In terms of constraints of the system, the earliest time of impact would be with

- the smallest angle of observation β_{min}.
- the fastest speed of the target ω_{max}.

• the maximum angle ψ_{max} which constrains the span of observation of the FO.

Let $t_{o\,min}$ denote the earliest firing time:

$$t_o = t_{obs} + \Delta\tau_1\,(\omega,\ \beta,\ t_{obs}) + \Delta\tau_3$$

then

$$t_{o\ min} = [\ K_1(\beta_{min},0) / \omega_{max}] + \Delta\tau_{2\ min}(\text{path \# 1})$$

(path #1 is the combination of links with the minimum delay)
with

$$K_1(\beta_{min},\ 0) = (y_t - y_{FO}) \cdot [(1/\tan(\psi_{max} - \beta_{min})) - (1/\tan\psi_{max})]$$

The earliest firing time corresponds to an impact time with an x-coordinate. This is specified by:

$$x_t = \omega \cdot t_{impact} = \omega \cdot t_o + \omega \cdot \Delta\tau_3 = \omega \cdot t_o + 2\omega\sqrt{[R/g]}\ \sin\alpha$$

From Figure D.2, the following equation can be made from the right triangle:

$$X^2 = (x_t - x_{Bi})^2 + (y_t - y_{Bi})^2$$

Substituting x_T into above yields:

$$\sin^2 2\alpha = \left[\frac{\omega t_o - x_{B1}}{R} + \frac{2\omega}{\sqrt{Rg}}\sin\alpha\right]^2 + \left[\frac{y_t - y_{B1}}{R}\right]^2$$

From the constants listed in Chapter 5 and the equations above, the numbers in Table D.1 are computed.

	Battery 1	Battery 2
α	21.0466°	12.23 °
$\Delta\tau_{3\,min}$	35.626 sec	21.0245 sec
α	21.60°	13.685 °
$\Delta\tau_{3\,max}$	36.512 sec	23.4775 sec

Table D.1. Time of Flight for Batteries

Because the time of flight is insensitive to the variation of the shooting angles ϕ_i , the times are set to constants to simplify the computations:

$\Delta\tau_3$ from $B_1 \Rightarrow$ 36 seconds.

$\Delta\tau_3$ from $B_2 \Rightarrow$ 22 seconds.

D.2. Derivation of the Uncertainty Expression $\rho = \delta\omega / \omega$

After the forward observer makes the first target measurement at t_{obs}, he waits until the target has moved an angle β to make the second observation. This causes a time delay of $\Delta\tau_1$ and then the estimate of the speed of the target and its position at t_{obs} is transmitted to to the central computer.

Figure D.3. shows the geometry of observation for a forward observer.

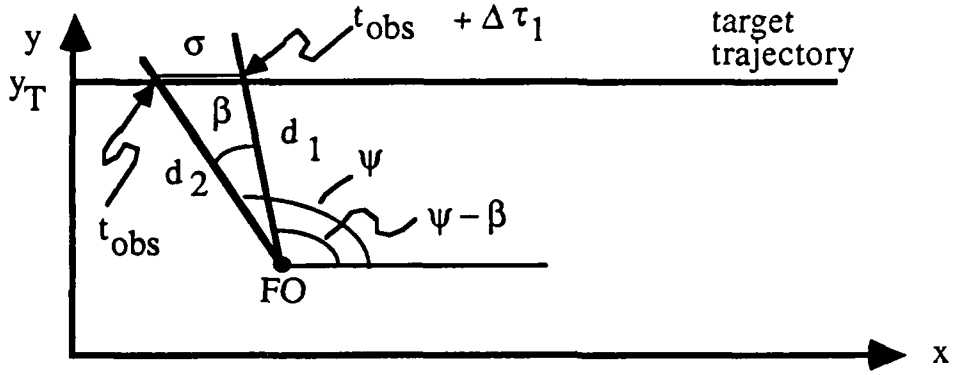


Figure D.3 Geometry of a Forward Observation

From Figure D.3, an expression for σ can be found using the law of cosines:

$$\sigma^2 = d_1^2 + d_2^2 - 2d_1d_2 \cos\beta \qquad \text{(D.12)}$$

Differentiating:

$$2\sigma\delta\sigma = 2d_1d_2 + 2d_2\delta d_2 - 2(d_1\delta d_2 + d_2\delta d_1)\cos\beta + 2d_1d_2\sin\beta\,\delta\beta \qquad \text{(D.13)}$$

where

$\delta d_1 = \delta d_2 = \delta d$ is the absolute error of the distance of measurement equipment of FO.

$\delta\beta$ is the absolute error in the reading of the angle β by FO.

δt is the absolute error in te timing equipment of FO.

For a small value of angle β:

$$d_1 \approx d_2 \approx d_o = 1/2\,(d_1 + d_2)$$

and substituting into equations (D.13) and (D.14) :

$$\sigma^2 = 2d_o^2 (1 - \cos \beta) \qquad (D.15)$$

$$\delta\sigma = d_o/ \sigma[2(1 - \cos \beta) \delta d + d_o \sin \beta \delta\beta] \qquad (D.16)$$

Finally,

$$\frac{\delta \sigma}{\sigma} = \frac{d_o \quad [\, 2\,(1- \cos\beta)\delta d \quad + d_o \sin \quad \beta\, \delta\beta \,]}{2\, d_o^2 (1 - \cos\beta)}$$

$$= \frac{\delta d}{d_o} + \frac{\delta\beta}{2 \tan\beta/2} \qquad (D.17)$$

The estimate of the speed of the target is specified by:

$$\omega = \sigma / \Delta \tau_1$$

and differentiating:

$$\delta\omega/\omega = \delta\sigma/\sigma \; + \delta (\Delta\tau_1)/ \Delta\tau_1 \; = \delta\sigma/\sigma \; + 0$$

The assumption is that there is no error in the timing equipment, $\delta t = 0$.
Substituting in Equation (D.16)

$$\rho = \delta\omega/\omega = (\delta d / d_o) + \; (\delta\beta / 2) \tan \beta/2$$

From Figure D.3, d_1 and d_2 can be specified:

$$d_1 = (y_t - y_{FO}) / \sin(\psi - \beta)$$

Therefore, the expression for uncertainty in the speed estimate is:

$$\rho = \delta\omega/\omega$$

$$= \frac{\delta d}{\frac{y_t - y_{FO}}{2}\left(\frac{1}{\sin\psi} + \frac{1}{\sin(\psi-\beta)}\right)} + \frac{\delta\beta}{2\tan\beta/2} \qquad \text{(D.18)}$$

D.3. Network Analysis

Figure D.4 shows the decision tree which represents the system operating scheme. Each of the links are considered and all the paths are taken. As demonstrated by the tree, there are only four possible paths that successful communication can occur and they are labeled path 1, 2, 3, and 4. Cothier (1984) goes into a detailed analysis for each of these paths in Appendix C of his Master's Thesis ("Network Analysis of TACFIRE", p. 95-100). He computes the time delay associated with each link's specific hardware and determines the overall time delay for commication along each path. These values are contained in the values of **v** and **u** described in Chapter 4.

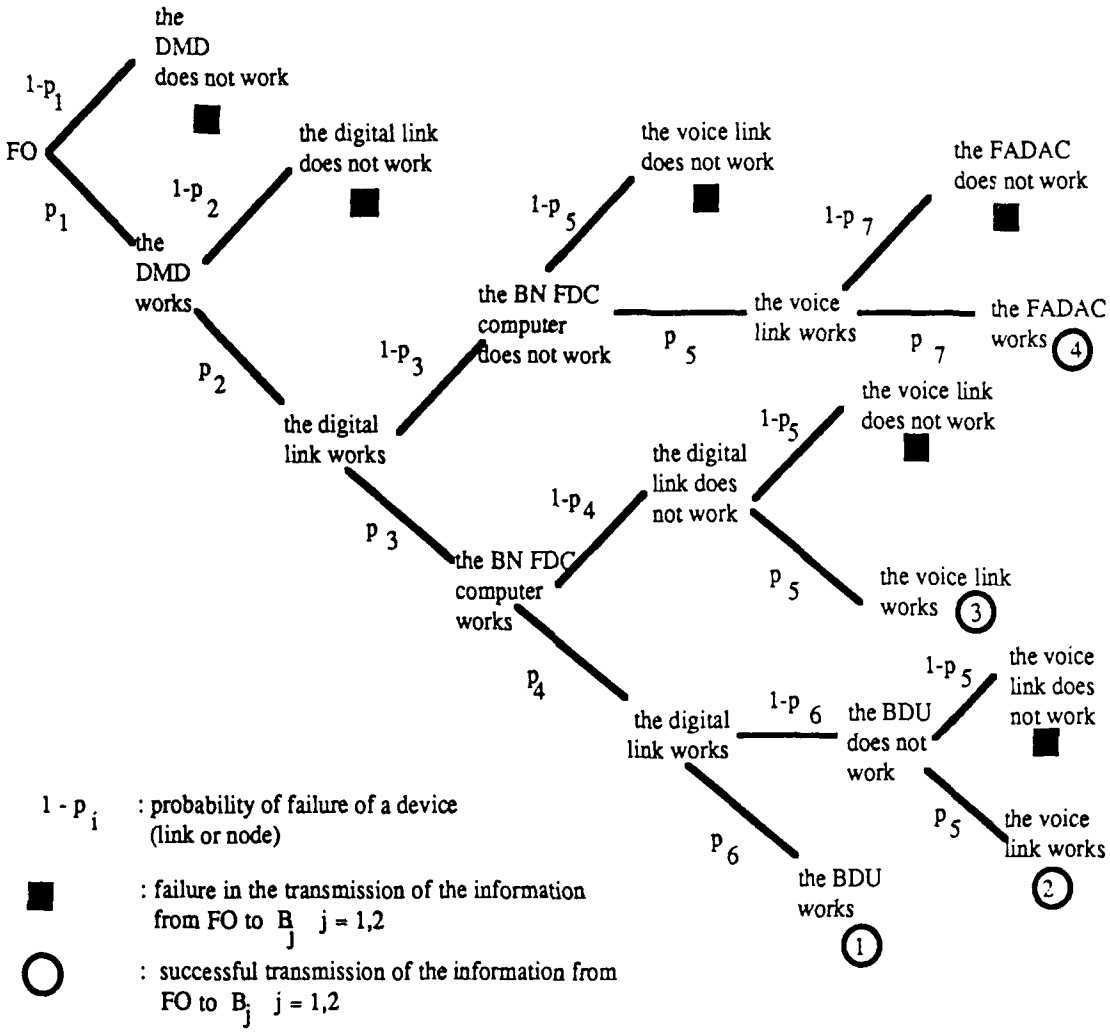


Figure D.4 Decision Tree Representing the System Operating Scheme

Cothier (1984) also derives the reliability /survivability of the network in Appendix D ("Reliability / Survivability analysis of the Network", p. 101 - 103) of his Master's Thesis. He finds that reliability / survivability is a function of the probabilities of the paths in which successful communication occurs.

$$\mathbf{RS(p)} = \sum_{i=1}^{4} q(i)$$

D. 4. Derivation of the System Attributes for the Two Battery Cases

The geometry of the two battery case is shown in Figure D.5 below.

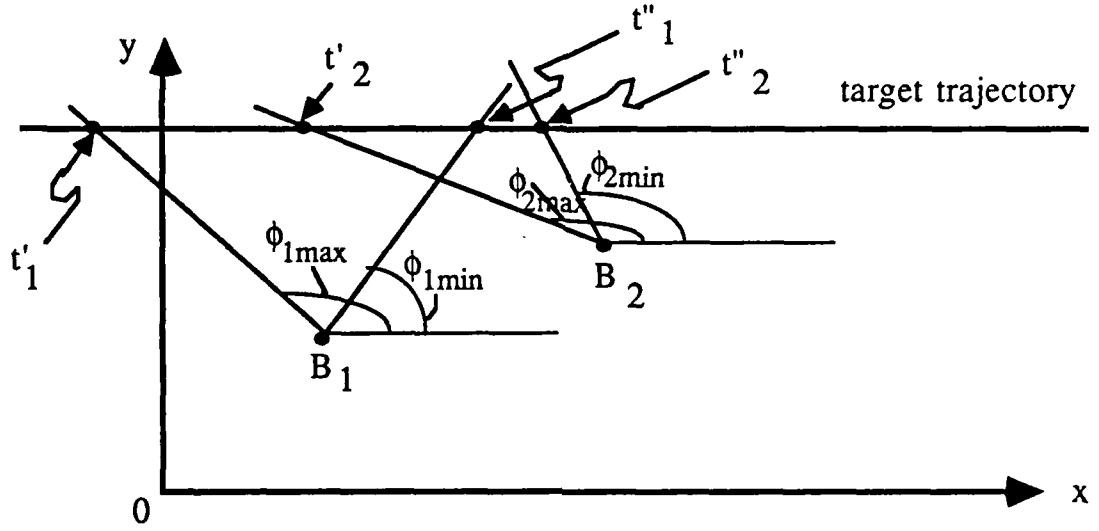


Figure D.5. The Two-Battery Case

Let:

$$K''_1 = x_{B1} + [y_T - y_{B1}] / \tan \phi_{1\ max}$$

$$K'_2 = x_{B2} + [y_T - y_{B2}] / \tan \phi_{2\ max},$$

$$K''_2 = x_{B2} + [y_T - y_{B2}] / \tan \phi_{2\ min}$$

Then, the times as marked in Figure D.5 are:

$$t''_1 = K''_1 / \omega$$

$$t'_2 = K'_2 / \omega, \qquad t''_2 = K''_2 / \omega$$

For simplicity, it is assumed that the paths from FO to the batteries are identical, that is, if path #3 is used to send information from FO to B_1 then path #3 is also used to send information from FO to B_2.

Option 1: no coordination

The window of opportunity is specified by the following equations:

$$t^{**} = t''_2 = K''_2 / \omega$$

$$\Delta t_i = K''_2 / \omega - \Delta\tau_1(\omega,\beta,t_{obs}) - u(i) \qquad \text{(D.19)}$$

and

$$\Delta t = \sum_{i=1}^{4} q(i) \cdot \Delta t_i \qquad \text{(D. 20)}$$

The Overall Probability of Kill for Doctrine 1 is developed by the following equations:

• For battery B_1

$$n_i = 1 + \text{int}\,(\Delta t_i / v(i)) \qquad \text{(D.21)}$$

and

$$OPK_i^{(B1)} = \prod_{n=0}^{n_i - 1} \frac{\xi(\omega,\beta,0)}{\Delta\tau(\omega,\beta,0) + u(i) + n \cdot v(i)} \qquad i = 1,...,4 \qquad \text{(D. 22)}$$

• For battery B_2, the shooting span is determined by $t''_2 - t'_2$. The number of possible shots is then

$$m_i = 1 + int[(t''_2 - t'_2)/v(i)] = 1 + int\ [\ (K''_2 - K'_2)/\omega \cdot v(i)] \qquad (D.23)$$

and

$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega,\beta,0)}{t'_2 + n \cdot v(i)} \qquad i = 1,...,4 \qquad (D.24)$$

or

$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega,\beta,0)}{\frac{K'_2}{\omega} + n \cdot v(i)} \qquad i = 1,...,4 \qquad (D.25)$$

For Doctrine 2, the number of shots is determined by the inequalities:

For battery B_1 -

$$t_{impact}(n_i) \le t''_2 < t_{impact}(n_{i+1})$$

with the first impact time $t_{impact}(1) = t'_1$

For battery B_2 -

$$t_{impact}(m_i) \le t''_2 < t_{impact}(m_{i+1})$$

with the first impact time $t_{impact}(1) = t'_2$

Because the impact times are also the next observation time, the following equations are used:

$$OPK_i^{(B_1)} = \prod_{n=0}^{n_i - 1} \frac{\xi(\omega,\beta,t_{obs}(n))}{t_{impact}(n) - t_{obs}(n)} \qquad i = 1,...,4 \qquad (D.26)$$

and

$$OPK_i^{(B2)} = \prod_{n=0}^{m_1 - 1} \frac{\xi(\omega, \beta, t_{obs}(n))}{t_{impact}(n) - t_{obs}(n)} \quad i = 1,...,4 \qquad (D.27)$$

The observation time for both batteries is the same for each n but the impact times are different.

When the OPK's are determined for each battery and path, the following equations are used to obtain the overall probability of kill:

$$OPK_i = 1 - (1 - OPK_i^{(B1)}) \cdot (1 - OPK_i^{(B2)}) \qquad (D.27)$$

$$OPK = \sum_{i=1}^{4} q(i) \cdot OPK_i \qquad (D.28)$$

Option 2: coordination

The window of opportunity is determined by the following equations:

$$t^{**} = t''_2 = K''_2 / \omega \qquad (D.29)$$

The width of the window of opportunity is determined by battery B_2 span of fire -

$$\Delta t_i = t''_2 - t'_2 = (K''_2 - K'_2)/\omega \qquad (D.30)$$

$$\Delta t = \sum_{i=1}^{4} q(i) \cdot \Delta t_i \qquad (D.31)$$

Observation Time

To compute the OPK's, the observation time must be computed - it is no longer at t= 0. The first impact time is t'_2 and assuming the worst case transmission (path #4 is the slowest), the

following equation can be made:

$$t_{obs} + \Delta\tau_1(\omega,\beta,t_{obs}) + \Delta\tau_2(\text{path \#4}) + \Delta\tau_3 = t'_2 \qquad \text{(D.31)}$$

The flight of the projectile is set to 36 seconds, which is the slower flight time from the two batteries.
Let

$$t_{end\ obs} = t_{obs} + \Delta\tau_1(\omega,\beta,t_{obs}) \qquad \text{(D.32)}$$

thus,

$$t_{end\ obs} = K'_2/\omega - \Delta\tau_{2\ min}(4) - \Delta\tau_3 \qquad \text{(D.33)}$$

From Figure D.3., the forward observation geometry,

$$\tan(\psi-\beta) = [y_t - y_{FO}] / [x_T(t_{end\ obs}) - x_{FO}]$$

where

$$x_T(t_{end\ obs}) = \omega \cdot t_{end\ obs}$$

Thus, ψ can be solved for by substituting equation (D.33) into the equations above:

$$\tan(\psi-\beta)=[y_t-y_{FO}]/[K'_2 - \omega\cdot\Delta\tau_{2\ min}(4) - \omega\cdot\Delta\tau_3 - x_{FO}] \qquad \text{(D.34)}$$

Now, $\Delta\tau_1$ can be found:

$$\Delta\tau_1 = \frac{y_T - y_{FO}}{\omega}\left(\frac{1}{\tan(\psi-\beta)} - \frac{1}{\tan\psi}\right) \qquad \text{(D.35)}$$

Lastly, t_{obs} is solved for by substituting equations (D.35) and (D.33) into equation (D.32):

$$t_{obs} = t_{end\ obs} - \Delta\tau_1 \qquad \text{(D.36)}$$

To determine the overall probability of kill, the number of shots from each battery must be determined. For Doctrine 1, the number of shots fired from battery B_1 is

$$t'_2 + (\eta_i - 1)\cdot v(i) \le t''_1 < t'_2 + \eta_i \cdot v(i) \qquad \text{(D.37)}$$

Thus,

$$OPK_i^{(B1)} = \prod_{n=0}^{\eta_i - 1} \frac{\xi(\omega, \beta, t_{obs})}{\frac{K'_2}{\omega} + n \cdot v(i) - t_{obs}} \qquad i = 1,...,4 \qquad \text{(D. 38)}$$

where t_{obs} is specified by equation (D.36).

For Doctrine 1, the number of shots from battery B_2 is given by:

$$m_i = 1 + int\ [(K''_2 - K'_2)/\ (\omega \cdot\ v(i))] \qquad \text{(D.39)}$$

and

$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega, \beta, t_{obs})}{\frac{K'_2}{\omega} + n \cdot v(i) - t_{obs}} \qquad i = 1,...,4 \qquad \text{(D. 40)}$$

For Doctrine 2, the number of shots from battery B_1 is determined by solving the inequality

$$t_{impact}(\eta_i) \le t''_1 < t_{impact}(\eta_{i+1}) \qquad \text{(D.41)}$$

and

$$OPK_i^{(B1)} = \prod_{n=0}^{\eta_i - 1} \frac{\xi(\omega, \beta, t_{obs}(n))}{t_{impact}(n) - t_{obs}(n)} \quad i = 1,...,4 \qquad (D.42)$$

and the number of shots from battery B_2 is determined by solving the inequality

$$t_{impact}(m_i) \leq t''_2 < t_{impact}(m_{i+1}) \qquad (D.43)$$

and

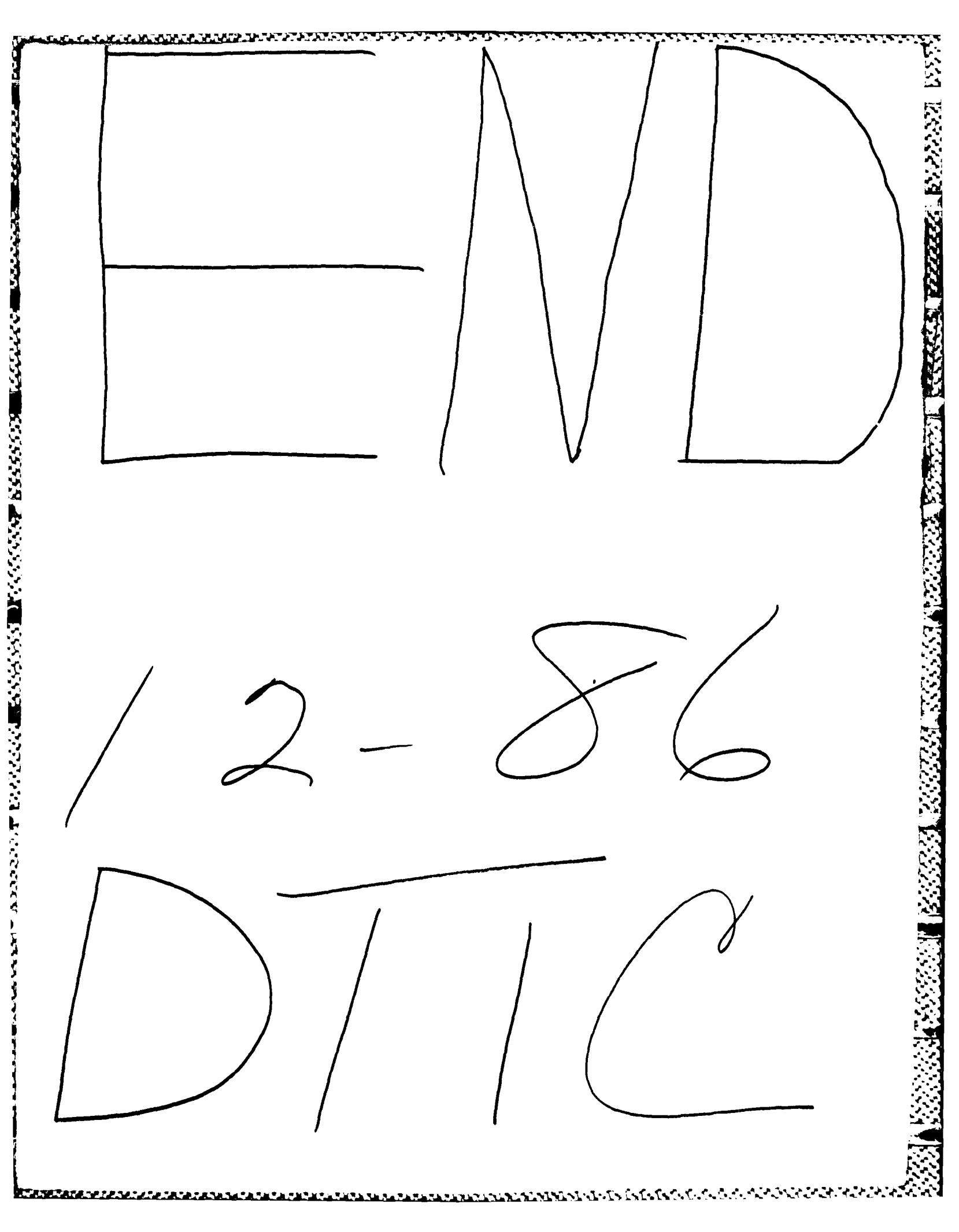
$$OPK_i^{(B2)} = \prod_{n=0}^{m_i - 1} \frac{\xi(\omega, \beta, t_{obs}(n))}{t_{impact}(n) - t_{obs}(n)} \quad i = 1,...,4 \qquad (D.44)$$

Finally,

$$OPK_i = 1 - (1 - OPK_i^{(B1)}) \cdot (1 - OPK_i^{(B2)}) \qquad (D.45)$$

$$OPK = \sum_{i=1}^{4} q(i) \cdot OPK_i \qquad (D.45)$$

The impact time for the battery B_2 are the same in both options.


END
12-86
DTIC